Zoo/PhytoImage version 1.2-0

Computer-Assisted Plankton Image Analysis

User's Manual

By

Philippe **Grosjean**

and

Kevin **Denis**

Table of content

Introduction

Planktonic organisms constitute the base of many aquatic food webs. Indeed zooplankton is the key mediator between energy synthesized by phytoplankton and higher trophic levels. Because plankton can vary quickly in term of abundance and biomass according to variations of environmental conditions, it constitutes a significant bio-indicator of global changes like increasing of atmospheric CO_2, global warming or anthropogenic eutrophication. The taxonomic composition, the distribution and the abundance of planktonic groups are thus fundamental parameters of ecosystem structures and mechanisms and can be used to understand and to quantify the contribution of these planktonic organisms to these processes. To understand these phenomena and because of the heterogeneity of plankton distribution it is necessary to increase spatial and temporal resolution of sampling method. In most of the case, the amount of samples can be quickly important. Unfortunately, the analysis of plankton samples is traditionally associated with long sessions of counting planktonic organisms under the binocular with formaldehyde vapours floating around. This limitation increases the time of samples treatment and constitutes the bottleneck of planktonology. Moreover, taxonomists are increasingly rare. Although this picture of a planktonologist will probably remain for a while, it seems to be another way to gather data about plankton: computer-assisted analysis of plankton digital images. This alternative and complementary method is considered since the 80's but was limited by the quality and the resolution of existing devices. Now, the combination of current powerful computers with the quality of digitalized devices and the efficiency of machine learning algorithms provide a potent tool to biologists and planktonologists. The computer-assisted analysis of plankton can provide rapid enumeration and identification of plankton samples. In order to help planktonologists in their works, a whole suite of hardwares to take pictures of plankton, both *in situ* and/or from fixed samples, is now developed: FlowCAM, laser OPC, VPR, Zooscan, … (more to come with Holocam, Sipper, Zoovis, HAB Buoy), not forgetting the use of a digital camera on top of a binocular or with a macro lens. Digital images of plankton are barely usable as such; they must be analyzed in a way that biologically and ecologically meaningful features are extracted from the pixels. But, all images digitalized by these hardwares need the intervention of taxonomists for plankton identification and classification. The machine learning method uses extracted features of some representative particles to automatically identify and classify all particles images in different taxonomical / ecological groups. A software doing such an analysis is thus an essential phase in digital image processing. From this perspective, different attempt were

made to process images of plankton in order to extract taxonomical and ecological information. DiCANN (Dinoflagellate Categorisation by Artificial Neural Network) is a software developed to analyze and classify dinoflagellates with neural network algorithms. This method discriminated 6 species with 72 % accuracy. REFLICS (REal FLow Imaging and Classification System) provides a software for the real-time automated recognition of fish eggs with few false negative by using nearest neighbours algorithm. The FlowCAM is provided with Visual Spreadsheet, a software which extracts features and creates size spectra of particles but can also recognize them according to their similitude. PVA (Plankton Visual Analyser) is a free software which counts, sizes and classifies particles of digital images (http://www.azti.es/), but the whole process is time consuming. More recently, the visual plankton software is developed to analyze images from VPR. Finally ZooProcess is a free software designed to process images from ZooScan but does not allow to automatically recognize particles (http://www.obs-vlfr.fr/LOV/ZooPart/ZooScan/).

All these softwares are specific to one or few image formats and usually specific to one digitisation device.

Zoo/PhytoImage aims to provide a powerful and feature-rich software. This software proposes a free solution to the use of plankton pictures from various origins and moreover, provides numerous table of measurements (i.e., abundances, total and partial size spectra, total and partial biomasses, …). Zoo/PhytoImage is not provided with any of the previously cited devices, and it is not going to be a commercial product. It is distributed for free (GPL license, distributed through the Zoo/PhytoImage web site, http://www.sciviews.org/Zoo/PhytoImage). Note that ZooImage and PhytoImage are essentially the same software, but with a different presentation and different scientists targeted (zoologists *versus* phycologists). It is an open source software. It means that it provides a general framework to import images, analyse them, and export results from and to a large number of systems. So, each image of plankton acquired from any device can be analysed by Zoo/PhytoImage and each developer can also contribute to the amelioration of the software. The Open Source approach of wiring many willing developers around the world in a common project has already shown its efficiency: Linux, Apache, but also R or ImageJ in the field of statistics and image analysis, respectively, are some examples. Like this, Zoo/PhytoImage is built from various existing software like ImageJ, R or XnView, and it provides itself new components, for the benefit of both users and developers. It runs currently on Windows but can also be easily adapted to Linuxes, various Unixes or Mac OSx (use of portable software and languages).

The general framework of Zoo/PhytoImage is designed to deal with images of various origins and characteristics. Consequently, it is not a streamlined and rigid system. It is rather made of a collection of different and customizable applications collected together into a single system. Currently, Zoo/PhytoImage can process images acquired in laboratory (fixed samples scanned with the FlowCAM, ZooScan for instance) and in the future, could analyse images from *in situ* devices like VPR or HAB Buoy.
This user's manual will guide you in your first use of Zoo/PhytoImage. Different examples are available with the software or downloadable on Zoo/PhytoImage web site to learn to use the software.

The current version of Zoo/PhytoImage is a "beta release". Make sure you test it completely before you use it in routine works. Also, take time to report possible bugs, and make suggestions to improve the software (or better, propose patch yourself!) This beta release is only currently working on Windows (other platforms will be supported later, as very little changes are required to support them). Also the complete documentation is not available yet ... meaning that most functionalities that are already there are not accessible yet, except if you carefully read the code to discover them yourself!
The present manual is presented for a specific use under Windows XP.

Quick overview of Zoo/PhytoImage

Zoo/PhytoImage is divided in three successive and logical sections: the image treatment, the automatic particles recognition and the analysis of temporal series.
The first part refers to images treatment and storage. Zoo/PhytoImage can work with high-resolution images and large quantity of data. For example a 16-bits grey image of 6000 x 10.000 pixels has yet a size of 120 Mb. With this amount of data, an inherent problem in the storage space is surging. It becomes obvious that images rapidly saturate your hard disk even if you want to work with a data base. Moreover, due to the need of large RAM memory, the process time is thus increased. Zoo/PhytoImage considers raw images as temporary data used for parameters extraction from the pixel information and not as final data. In fact, Zoo/PhytoImage decreases information weight from large size raw images into compressed files. For example, each sample composed of 16-bits raw image weights **720** Mb (6 raw images of 120 Mb each one) is compressed by the software into a file of **20** Mb. These compressed files, called "zid" files in **Z**oo/Phyto**I**mage **D**ata, contain the useful data for the further sections of the analysis (the automatic recognition and the analysis of whole series).

After this image processing, raw images and metadata associated can be stored on external support waiting to be analysed with a further version of Zoo/PhytoImage. Only zid files are stored on your computer hard disk for the rest of analysis.

The second section of Zoo/PhytoImage focuses on machine learning and recognition tool creation. Once your images analysed, the software does not have yet the ability to automatically recognise and classify particles in groups of interest (copepods or diatoms…). In that optic, a training set (manual classification of representative particles in different groups) has to be created. A particularity of Zoo/PhytoImage training set is its hierarchical structure. Indeed, the training set is constituted by directories and subdirectories which respect taxonomical filiations. These filiations are necessary in our context because they can represent different taxonomical or ecological details, information useful for a spatial or temporal analyse of your samples. This structure allows a high flexibility because groups are not static and the hierarchy by pooling or dividing groups can easily be modified according to your aims. The software currently proposes various common machine learning algorithms but other one could be added in the software (R package). The recognition tool created by the learning phase can be analysed to assess its efficiency. Actually the confusion matrix which shows error between manual and automatic classification is available to optimise the efficiency of automatic recognition. When your recognition tool is adequate, a back up is proposed in order to load it directly for a further use of the software. Because it is a time consuming step particularly for large training set, Zoo/PhytoImage advise you to save it on your computer.

The third and last section focuses on biological and ecological analysis of the temporal series. This step uses the zid files and recognition tools to automatically recognise all particles and extract results like abundances, biomasses and size spectra from samples. Your sample can be split in different size fractions but also each fraction can have various replicates. Zoo/PhytoImage is designed to take in consideration all this complexity. All this part is fully automated and is designed to analyse complex series. This section is definitely suitable for processing information in batch; it means that the software automatically analyses series image by image or zid file by zid file. It allows you to analyse a large quantity of data without never saturate of your RAM memory. Moreover, at each step of analyse a log file is displayed and gives information about the current action. If errors occur, the process continues and a message appears in log file in order to target the problem and give you useful information to apply your correction.

Zoo/PhytoImage can be seen as a tool box under a graphic interface based on R and ImageJ, two free software. Image J is a powerful image analysis software distributed as freeware licence and written in Java language. A lot of different improvements can be made on ImageJ like new features to measure or new plug-ins to implement. R is also extremely customisable, new machine learning algorithms can be implemented by downloading other libraries of functions but it is also a powerful statistical software. Indeed, R has a huge capacity of data analysis and graphic presentation. In this context, the Matlab alternative is not a solution because it is not a free software.

Installation

1. Hardware requirement

Image analysis and automatic classification of images are computer-intensive processes, and you will likely analyze lots of objects (typically, hundreds of thousands, or millions of them). Thus, you need a recent and powerful computer to run Zoo/PhytoImage decently.

- Computer with a Pentium IV 3 Ghz or better processor.
- Minimum 512 Mb of RAM memory. Depending on the size of the images you want to analyze, you may need even more. Very large images (40 gigas or more) require at least 2Gb of RAM.
- **Adapted graphic card and screen**. Chose a rapid, optimized graphic card capable of displaying 1280×1024, or 1600×1200 pixels with 24/32-bit color depth (millions of colors), associated with a high quality screen of no less than 17'. A screen of 19' or even 20' is not a luxury in using Zoo/PhytoImage!
- Although Zoo/PhytoImage optimizes disk space by compressing all files, dealing with lots of high-resolution pictures is consuming a lot of space on disk. You will need a **fast hard disk of at least 200 Gb of capacity**.
- Finally, a good **backup system** is also required. The later one can be made of large external hard disks of 200 or 300 Gb each, and/or of a fast DVD burner.

2. Download of the software

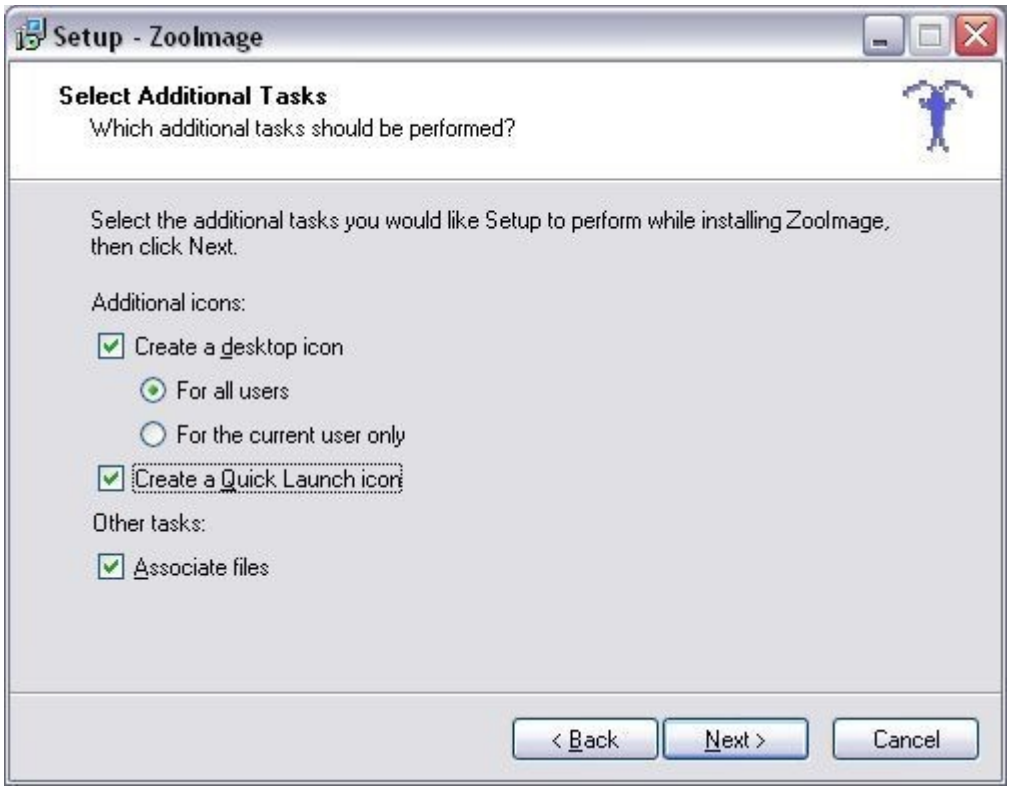
The software is available for download on the Zoo/PhytoImage website (http://www.sciviews.org/Zoo/PhytoImage/). Always remind that ZooImage and PhytoImage are indeed the same software, but with a different aspect… and targets different planktonologists!

3. Installation of Zoo/PhytoImage / PhytoImage under Windows

Zoo/PhytoImage will use about 400 Mo of space on your hard disk, when installed. You just have to execute the "**Zoo/PhytoImage _[x.y-z]Setup.exe**" file that you downloaded and to follow the installer's instructions step-by-step. Default values for the options should be fine, if you don't understand them.

The following screenshots shows how to install Zoo/PhytoImage.

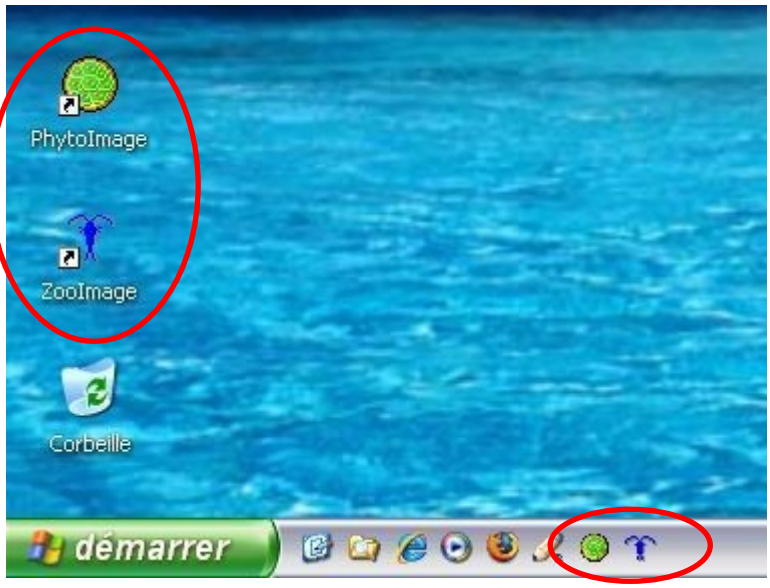
The first screen of the Zoo/PhytoImage installation assistant.



You can create desktop and quick launch icon (in the quick launch bar).

*It is very important to **associate files** with Zoo/PhytoImage: those files have special extensions and it will not be possible to open them by a double-click in the Windows explorer if you don't select this option. So, leave this option checked unless you have good reason to change it!*

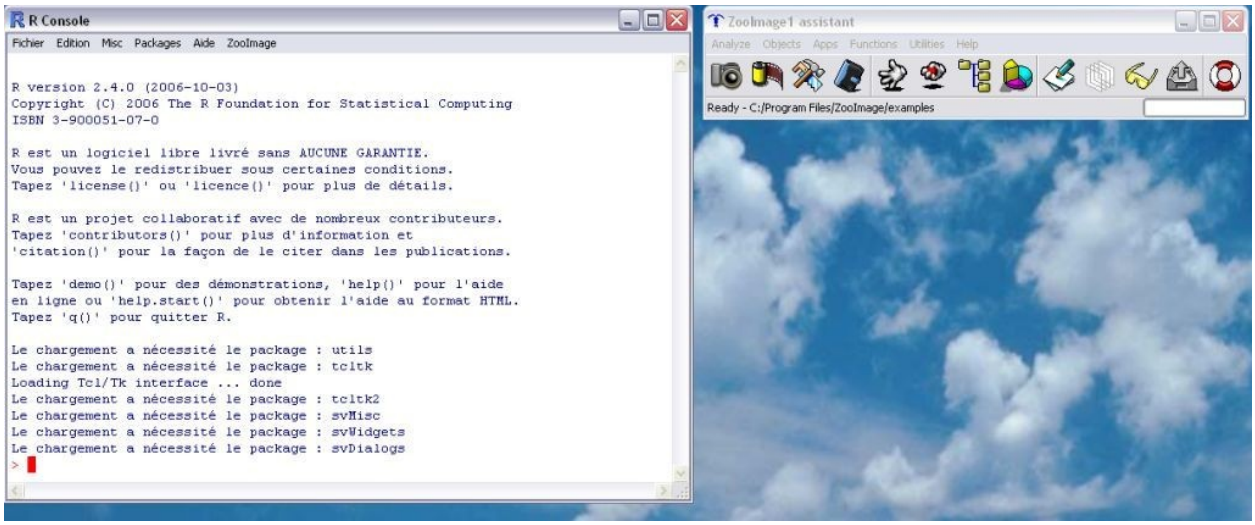
At the end of the installation, you should have ZooImage and PhytoImage entries in the start menu, and possibly ZooImage and PhytoImage icons on your desktop (if you left that option checked), one for starting the program as ZooImage, and the second one to start it as PhytoImage.



Example of desktop with ZooImage and PhytoImage icons (a copepod and a diatom respectively).

First use of Zoo/PhytoImage

When you double-click on the Zoo/PhytoImage icon on the desktop, or select the Zoo/PhytoImage (R) entry in the start menu, two windows appear on screen: the **R console** and the **Zoo/PhytoImage 1 assistant**.



When you first start Zoo/PhytoImage, you have two windows: one is called 'R Console'. It is the interactive window of R. The second one is named 'Zoo/PhytoImage 1 assistant'. The "Zoo/PhytoImage 1 assistant" toolbar is the interface between the user and the software. (It is with its menu and toolbar that you can easily use most of Zoo/PhytoImage functionalities).

The **R console** allows you to control R directly through command lines. You should not worry about this window, unless you are familiar with the R language. However, it logs important results and messages from your actions in Zoo/PhytoImage.

The **Zoo/PhytoImage 1 assistant** window is a toolbox with a menu on top and a status bar on bottom. It will guide you during the whole process. Basically, you just have to click on the buttons from left to right to run the various steps of your analysis.

A Zoo/PhytoImage analysis is subdivided in three parts. For each part, you have four buttons:

The three parts of the Zoo/PhytoImage process*, materialized by three times four buttons. The last button give help* (this Zoo/PhytoImage user's *manual*).

- The **first section** deals with image importation and process.

1. **Acquire images**. Start an external acquisition software (Vuescan, or any other program).
2. **Import existing images**. Possibly convert the format of the images and/or rename them. If images are already in correct format, this function just makes sure they have suitable *metadata* associated.
3. **Process images**. Basically, ImageJ is started. You are supposed to use one of the Zoo/PhytoImage -specific plug-ins in ImageJ to process your pictures.
4. **Make .zid files**. 'Zid' files stands for 'Zoo/PhytoImage Data' files. They contain all you need for the rest of the treatment, i.e., images of each individual (vignettes), their measurements and the metadata. Yet, they store this information in a compressed way.

- The **second section** helps you to make an automatic classifier optimized for your plankton series.

1.  **Make a training set**. This function prepares a directory with a hierarchy of subdirectories representing your manual classification (you can freely modify this structure like you want) and extract vignettes from the samples you think representative of annual variability. Then, you have to classify vignettes manually on the screen by moving them with the mouse to their respective group directory.
2.  **Read training set**. Once you have manually sorted the vignettes, this function collects this information into Zoo/PhytoImage. Statistics about your classification (number of vignettes in each group) is the displayed.
3.  **Make classifier**. Use a manual training to train an automatic classifier. You can use various algorithms of classification. You will get statistical results at the end of the process to evaluate performances of your classifier (cross-validation).
4.  **Analyze classifier**. This function analyses your classifier's performances. Currently, only the confusion matrix showing differences between manual and automatic classification, is calculated. Other diagnostic tools will be added in future versions.

- The **third section** uses this classifier and the measurements done on all objects identified in your pictures (first part) to calculate automatically abundances, biomasses and size spectra in all your samples. You can then visualize results, or export them.

1.  **Edit samples description**. Planktonic time series are identified by a list written in a specific Zoo/PhytoImage format. This list contains also further metadata about the series, and you have the opportunity to append various other measurements to the samples data (temperature, salinity, fluorescence, etc.).
2.  **Process samples**. This is the workhorse function that processes samples one by one for a given series, (1) identifying all individuals using your automatic classifier, (2) computing abundances, (3) calculating size classes for size spectra representations and studies, and (4) computing biomasses, using a table of conversion from total individual length or body area or ECD (Equivalent Circular Diameter) to carbon content, dry weight, etc. All these parameters are calculated for the whole sample but

also for each taxa and are converted per m³ of sea water, if suitable 'dilution' information is available in the metadata.

3.  **View results**. This option graphically presents results. You can draw composite exploratory graphs (up to 12 different graphs on the same page), even time series of abundances or biomasses changes, or size spectra of given samples.

4.  **Export results**. Results are written on the hard disk in ASCII format. This format is readable by any other software (Excel, Matlab, etc.).

Although you can export your results to analyze them in different software, you don't ***have*** *to do so. Zoo/PhytoImage operates in a R session and thousands of R functions are available for producing even the most sophisticated statistical analyses and graphs without leaving Zoo/PhytoImage-R.*

-  **Manual**. Display this PDF version of the user's manual.

Because the best way to approach to Zoo/PhytoImage software is to use it with training examples. You first have to download examples from the Zoo/PhytoImage web site. Each data game contains raw images, metadata associated with your images (acquisition device, color images, pixel resolution, sample number, ...) or training set and zid files which allow you to have an idea of the Zoo/PhytoImage performances for automatic recognition and series analysis.

All examples are available on Zoo/PhytoImage web site (http://www.sciviews.org/zooimage). We propose you different samples acquired from various devices (microscope images, macro photography, scanned samples, FlowCAM, ...). Some samples have too few images to perform all the analysis process. That is why we present below a table with the different samples and the steps of the software until were you can run the analysis. This is not due to the limitation of the software but only to the poor number of samples in some case.

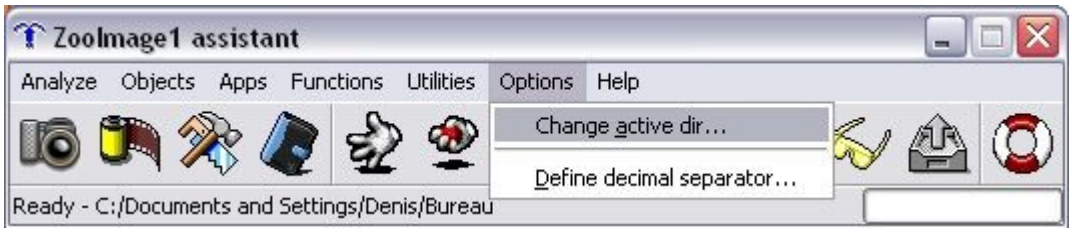
Table of available Zoo/PhytoImage examples. Grey cases represent parts of Zoo/PhytoImage treatment you may realize.

Examples	***Images treatment***	***Automatic recognition***	***Analysis of series***
FlowCAM-examples			
MacroG16-example			
MacroG16-train&data			

ScanCol24-example			
ScanCol24-train&data			
ScanG16-example			
ScanG16-train&data			
MicroCol24-example			

Because the software needs to know if all the images can be processed in a same way, metadata files are essential if you want to process a series of images in batch. This information is very important when you want to share yours images with an other laboratory or only to compare results.

When you downloaded one example, you can select it as the active one. Indeed, Zoo/PhytoImage always starts from the current active directory when you have to browse for files and subdirectories, you will gain time to switch it to the one where you store your raw images. The active directory is displayed in the status bar of the Zoo/PhytoImage 1 assistant window. To change it, use the Options → Change active dir... menu entry.



All files created by the software like _raw, _work and _train directories, will be located in the active directory. Notice that these directories begin by an underscore. In Zoo/PhytoImage, it means that none image will be processed if the underscore exists in front of the directory's name. It is a kind of protection to avoid a double processing. So the _raw directory will contain only your initial raw images and metadata associated and the _work directory will contain images created during the image analysis. These last images are only used for a visual approach to check if all the steps of image processing have been well executed. The _train directory will contain the vignettes manually classified in a hierarchy structure of directories and sub-directories. This classification is used by machine learning algorithms to automatically recognise plankton particles.

Section I: Image acquisition and analysis

The first section of Zoo/PhytoImage analysis focus on all steps referring to images acquisition and analysis. The software considers images like temporary information because the final goal is the extraction of biological and ecological information from images of samples. In this

sense, Zoo/PhytoImage will compress useful information from raw images to files used for the rest of the analysis.

1. Acquire digital images of plankton

1.1. Digitalised plankton samples in a goal of image analysis

Before starting to acquire images of your samples from any kind of device, you firstly need to know exactly the purpose of the images analysis. It means to determine until which level of particle recognition you want to arrive at (Species, Family, Class..). Even it is better to digitize particles with the maximum resolution, it is not always necessary, in fact taxonomical resolution depends on your pixel size resolution. Indeed, a detailed classification of your plankton will usually be associated with a high pixel resolution. Various devices exist to capture your plankton from fresh and fixed sample:

- ZooScan / flatbed scanner principle: Your sample is placed on the scanner window and then digitised. This technique allows you to obtain images with high resolution, for example 16 bits grey images at 2400 dpi (dot per inch) or more.
- Macrophotography: Your samples are digitised with a macro objective under an underlying lighting. This technique is an alternative to the scanner principle but for the same digitised surface you have to take more pictures. This technique is more difficult to carry out.
- Microphotography: Your samples are digitised with a camera located on a microscope or binocular. It is an easy to use method which gives high resolution pictures but usually limited by the number of particles on each image.
- Images acquired with a flow-cytometry based hardware like FlowCAM. This high-resolution method can be used on fixed or fresh samples. Particles are digitalised in a flow of water and vignettes of them are stored on computer. With new version of this device, you can also analyse mesoplankton (http://www.fluidimaging.com).
- Images acquired with *in situ* devices like VPR (Video Plankton Recorder) (http://www.whoi.edu/page.do?pid=11348) or SIPPER (Shadowed Image Particle Profiling Evaluation Recorder (http://www.marine.usf.edu/sipper/): these methods use high-speed digital camera to continuously sample plankton and suspended particles in a water column. Region of interest (ROI) are detected and stored on computer.

Functions of the taxonomical resolution, you have to choose the resolution of your acquisition device and your images format type. Indeed, if you want to study species, you have to choose a non-compressed images format like TIFF and a high resolution (1 pixel ~ 10 µm or smaller for micro and mesozooplankton). But if you want to study general groups like copepods or ecological groups, you can save your images as compressed format like JPEG and choose lower resolution. Pay attention, in each case, you have to save raw images without any modifications by the digitisation software in order to have raw signal directly from your hardware, except the compression of image format (JPEG for instance).

Zoo/PhytoImage is **not** a digitizing software. It is only designed to analyze existing digital images. However, for convenience, it binds to your favourite external acquisition software. As an example, if you use a digital camera with a dedicated capture software, you can specify the software in Zoo/PhytoImage and start it from the Zoo/PhytoImage 1 assistant in one click. You just have to select ".exe" file of your acquisition software.

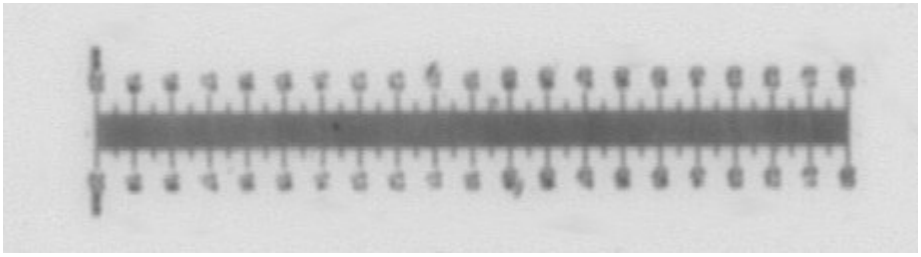
In Zoo/PhytoImage, you have the choice between two options to acquire your plankton images: Vuescan (test version available with the software) or your favourite acquisition software.

Zoo/PhytoImage is bundled with **Vuescan**, an excellent and very powerful software to acquire pictures from more than 400 commercial flatbed scanners and from more than 100 different RAW formats of digital cameras. Here we explain how to use Vuescan with a flatbed scanner for calibration to get digital plankton images… **but it should be clear that it is just an example: you are free to use any hardware/software combination you like to acquire your images!** Vuescan was used to obtain raw images from the ScanG16-example.

Vuescan is not a free software. It is a shareware distributed in two versions: personal and professional. You need the professional version. Its license is about $89, and you have to register your license with the author of Vuescan (see instruction in the Vuescan online help). We got the right to redistribute the trial version with Zoo/PhytoImage, but you have to unleash full features by entering your license code before using it in production.

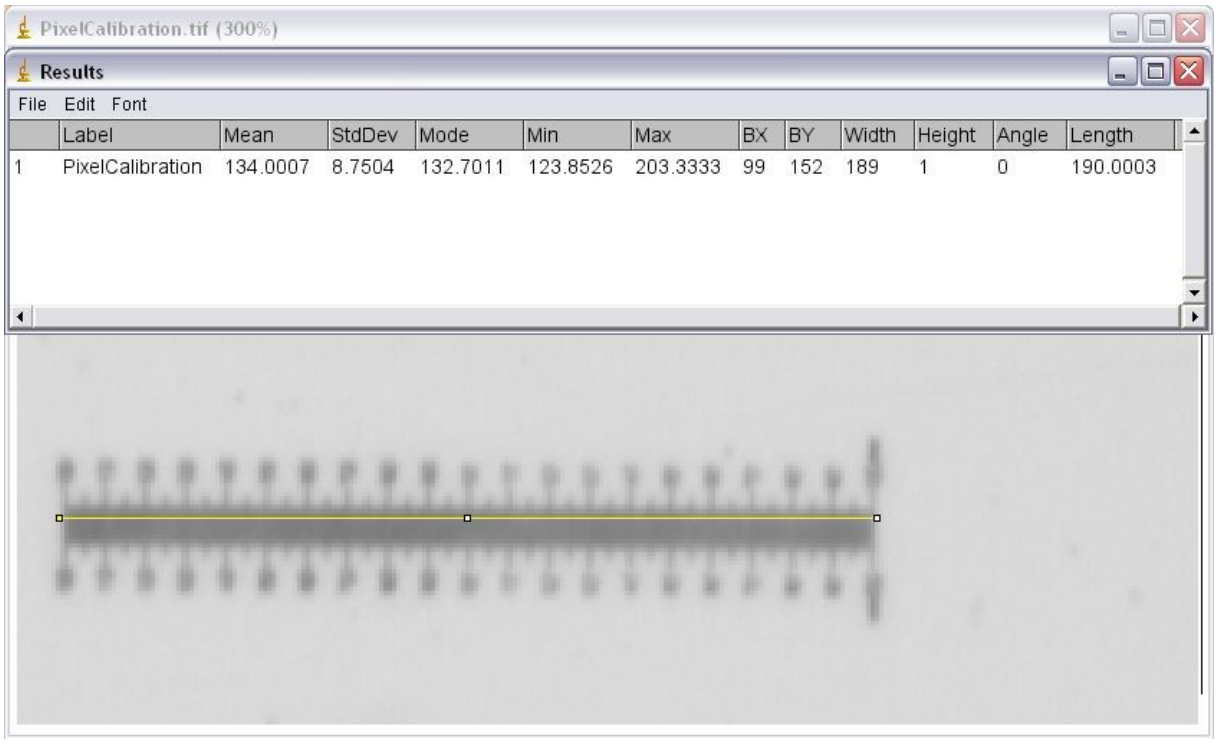
When you have determined the best compromise between the pixel resolution and the level of taxonomical recognition, you can calibrate your raw images spatially and in grey levels.

In order to quantitatively measure metric parameters of particles, you have to exactly calibrate the pixel size of your images. The digitalization of a micrograduate ruler or microspherical beads seem to be a good solution.



Raw 16-bits grey image of 2mm standard microscopic ruler digitalised at 2400 dpi on a flatbed scanner.

When the optimal resolution is determined (for example 2400 dpi), the real pixel size and its invariance have to be checked on all surface of digitization. First, you have to digitise a micrograduate ruler and measure its length.



Calibration of pixel size in ImageJ by the measurement of micrograduate ruler length. The 190 pixels correspond to a length of 2 mm, and thus 1 pixel = 10.53 µm.

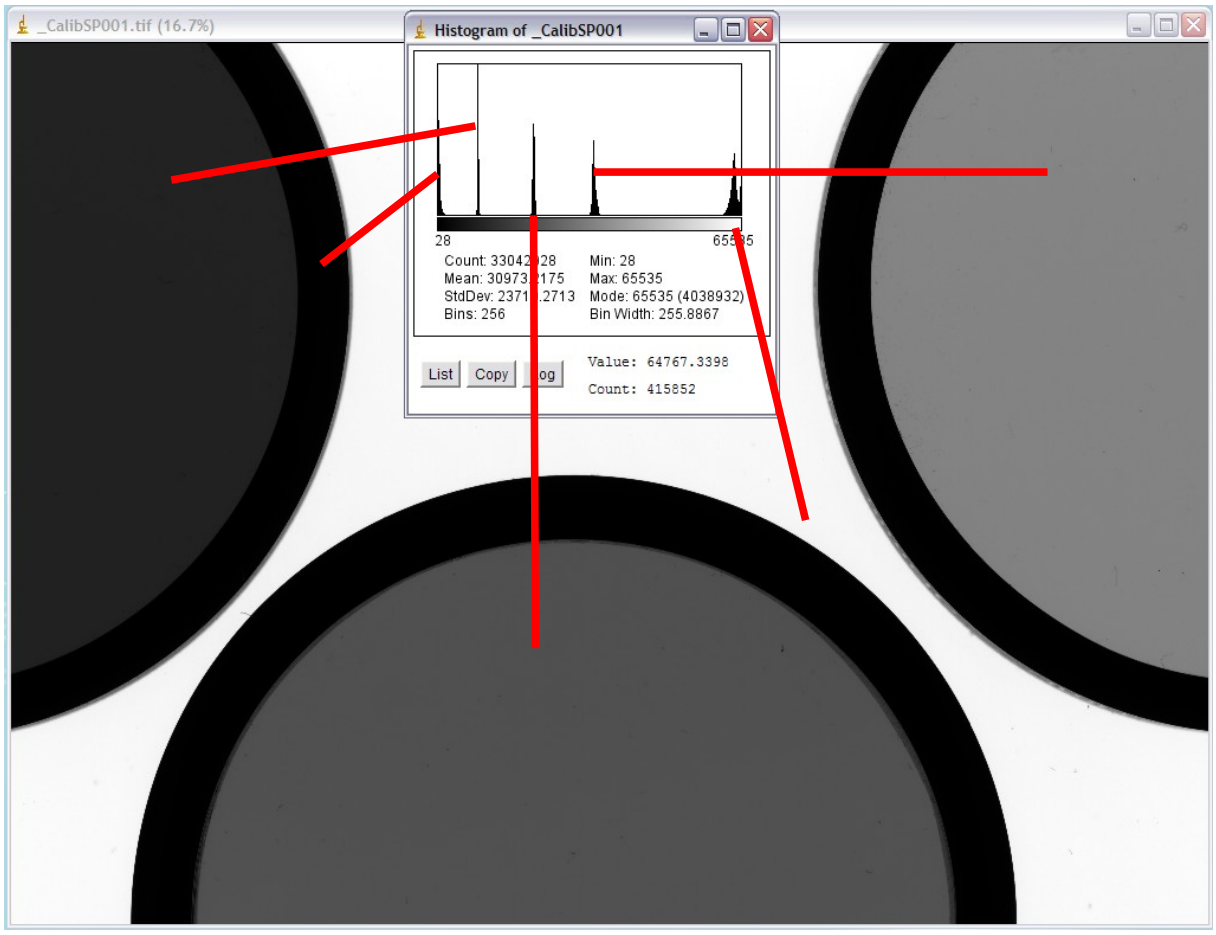
Knowing length of the micrograduate ruler, you can easily determine the pixel size. This operation has to be done with scale pointed horizontally and vertically to check the invariance of pixel size. This step is very important because the precision of your image analysis depend on this parameter.

To use parameters linked to grey levels in order to discriminate particles, you need to calibrate the grey levels of your scanner. To quantify grey levels in digitised particles, you have to verify the linearity of the captor as well as the correspondence between grey levels and transparency. The transparency can be measured in term of transmittance or Optical Density (O.D). The O.D. is the logarithm of base 10 of the reverse of transmittance, a transmittance of 0.25 means that object attenuates the light 4 times.

The method used for exactly scaling the range of grey level is quite easy. Three grey filters “Hoya type HMC” (NDx2 to NDx8) are digitized in the same image. These filters have a fixed transmittance of 0.5, 0.25 and 0.125 (either a beam attenuation of 2, 4 and 8 respectively).

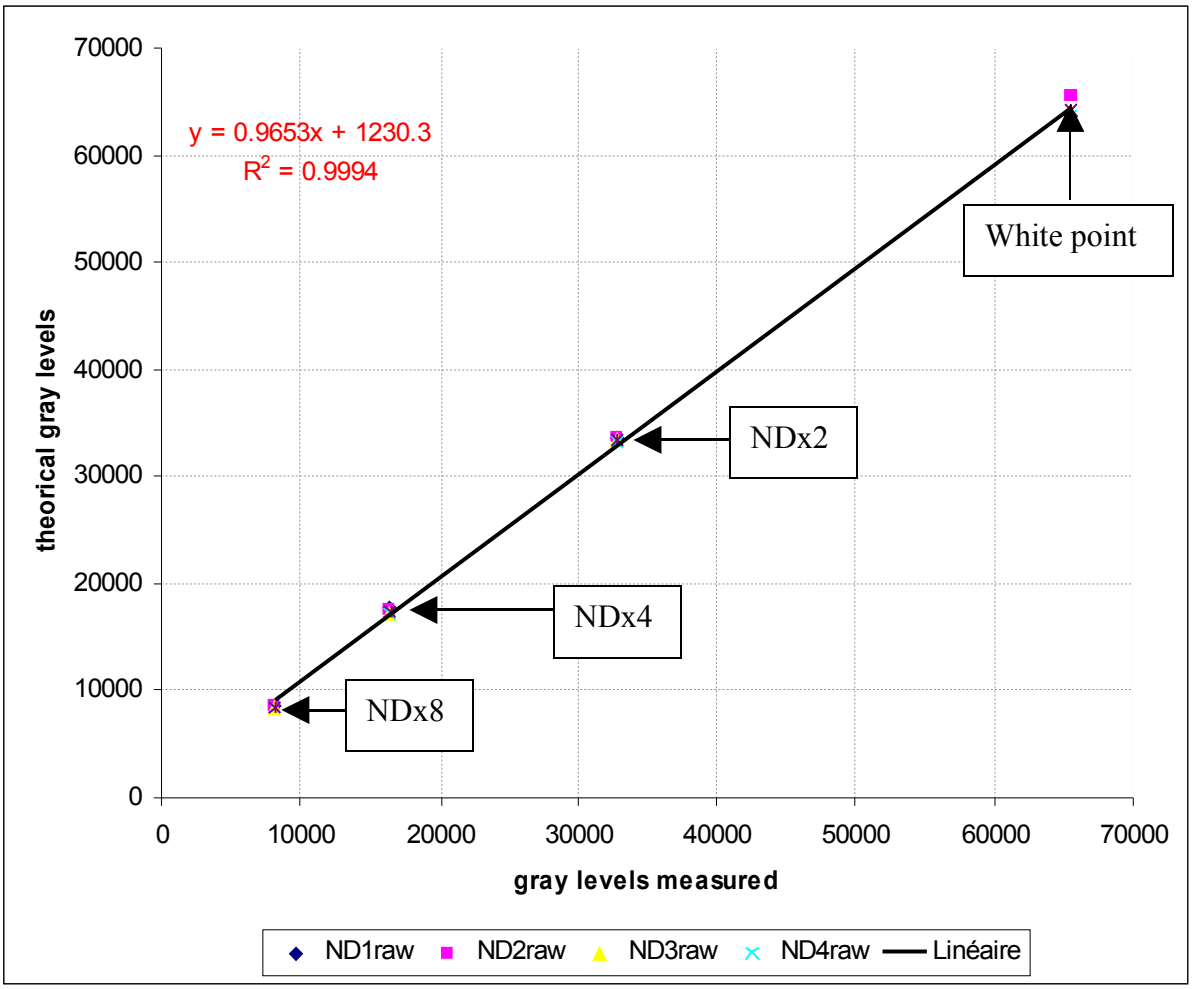
The grey filters Hoya NDx2, NDx4 and NDx8 which decrease the light intensity by 2, 4 and 8 respectively.

To determine the linearity of the signal, three grey filters Hoya type HMC are digitalized various times (every 2 hours) to evaluate possible change in the signal during on day. The grey levels of the three filters and the bottom (white) are measured with ImageJ



Measurement in ImageJ of grey levels on 16-bits raw image with grey filters. The modes of each filter (see graph) are used for the automatic calibration made by Zoo/PhytoImage.

A linear regression is then applied in order to extract the parameters of the relationship.



Calibration of grey levels in transmittance on raw images (16 bits, 2400 dpi). ND1 to ND4 are the four repetitions of measurements. We observe the linearity of the scanner captor, it means there are no transformations applied during digitization.

The regression equation is:

$T = 0.965\ R + 1230$

where T is the transmittance;

R the grey level of the scanner.

We have just checked the linearity of the response in term of transmittance. So, we conclude that the grey level of our scanner did not change during the time of use.

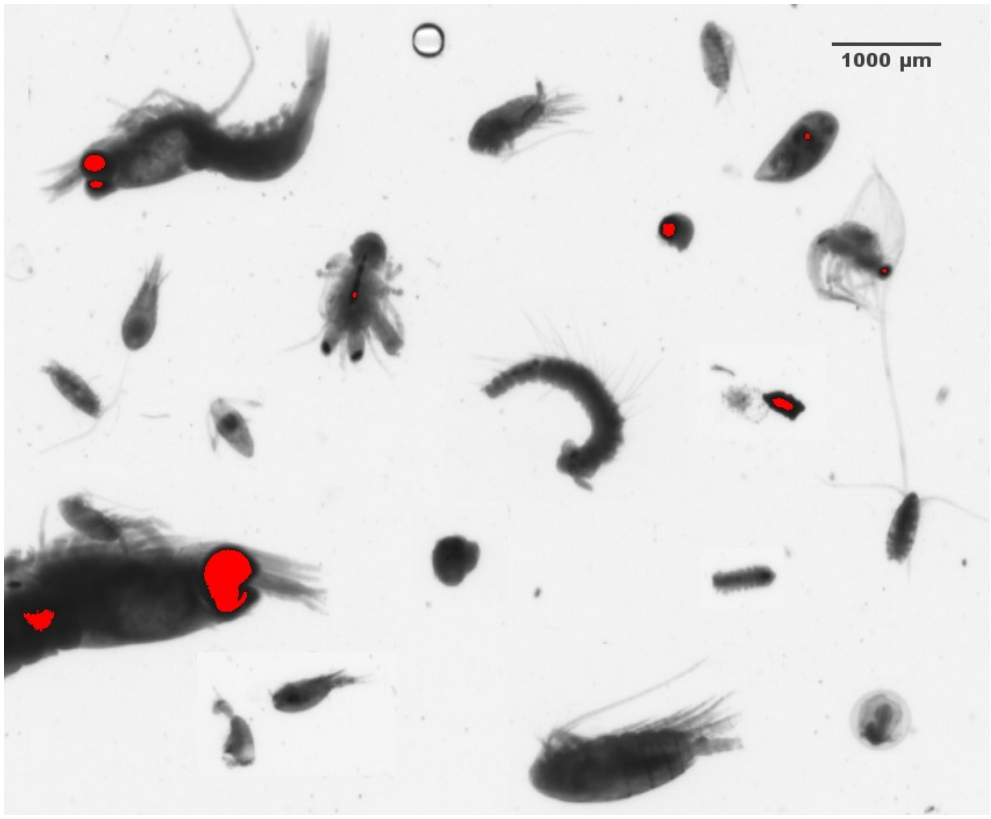
An other important point to consider before starting to digitize your samples is the subtraction of the background. In fact, an inherent noise associated to the pixels corresponding to the background (water and bottom of digitization cell) is always present. To remove this noise in

order to obtain the most homogeneous background and only measure grey levels of particles, the transmittance can not be used. Indeed, grey levels of one particle before and after background subtraction do not correspond in term of transmittance. Fortunately, grey levels can be measured in term of optical density. For example an O.D. of 0 corresponds to a transmittance of 100% and a transmittance of 10% corresponds to an O.D. of 1. An interesting property of the O.D. is the additivity of grey levels. Because of this O.D. property, its possible to subtract effect of the background to obtain the real O.D. of the particles of interest with the minimum noise.

The O.D. transformation is done automatically as follow:

- Calibration of the O.D. grey levels (using Hoya filters);
- Conversion of raw image from 16-bits to 8-bits of grey levels;
- Rescaling of the O.D. between 0 and 1 in the available grey levels (256 for a 8-bits grey levels);
- Inversion of image: usually black is coded as 0 and white as 255 in a 8 bits images but a negative image with white particles on black background is proportional to the O.D., so 0 for background and increasing of O.D. according to their opacity.

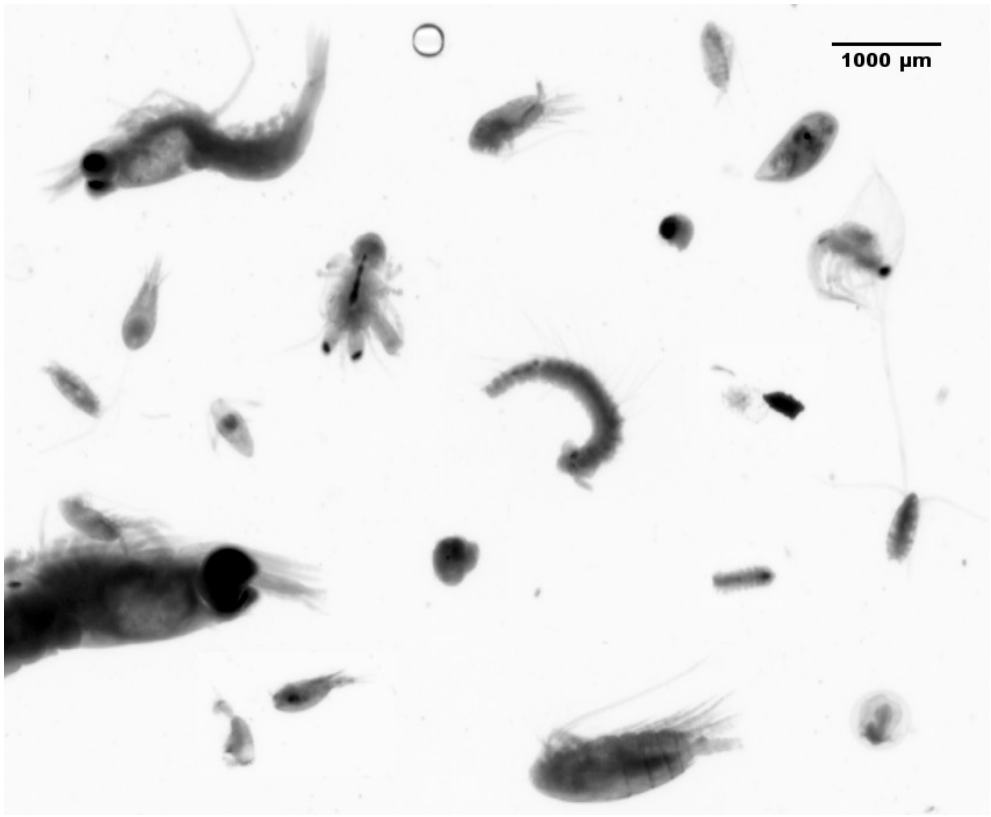
The 3 first operations are integrated in one step: calibration where the black and white points are calculated using modes of grey filters.



As you can see on this figure, parts of the 16-bits grey image having a O.D. higher than 1 (transmittance inferior to 10%) are very limited (red zones).

This figure represents a part of a 16-bits grey image having a O.D. higher than 1 (transmittance inferior to 10%) are very limited to the darkest zones.

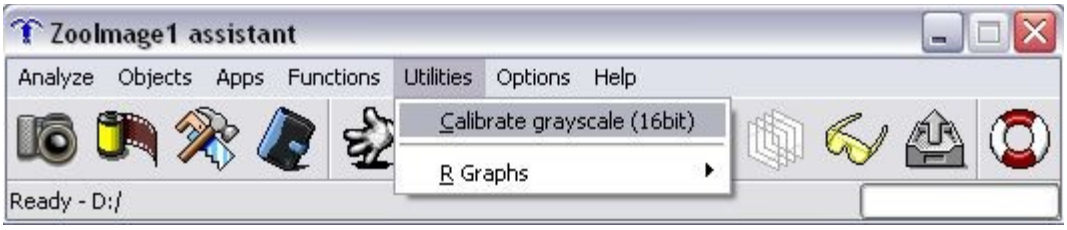
So we may limit the O.D range value and rescale them between 0 and 1 without loosing information on grey levels during transformation in 8-bits. Moreover, this D.O transformation permits to reduce the weight of the image.



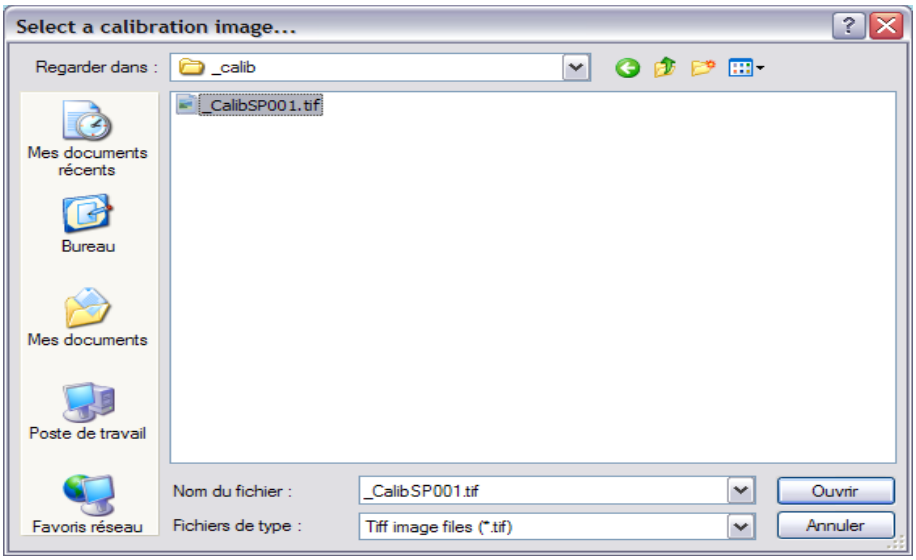
Same image after transformation into 8-bits and scaling of the O.D. between 0 and 1. Parts of the image having an O.D. superior to 1 are compressed in black.

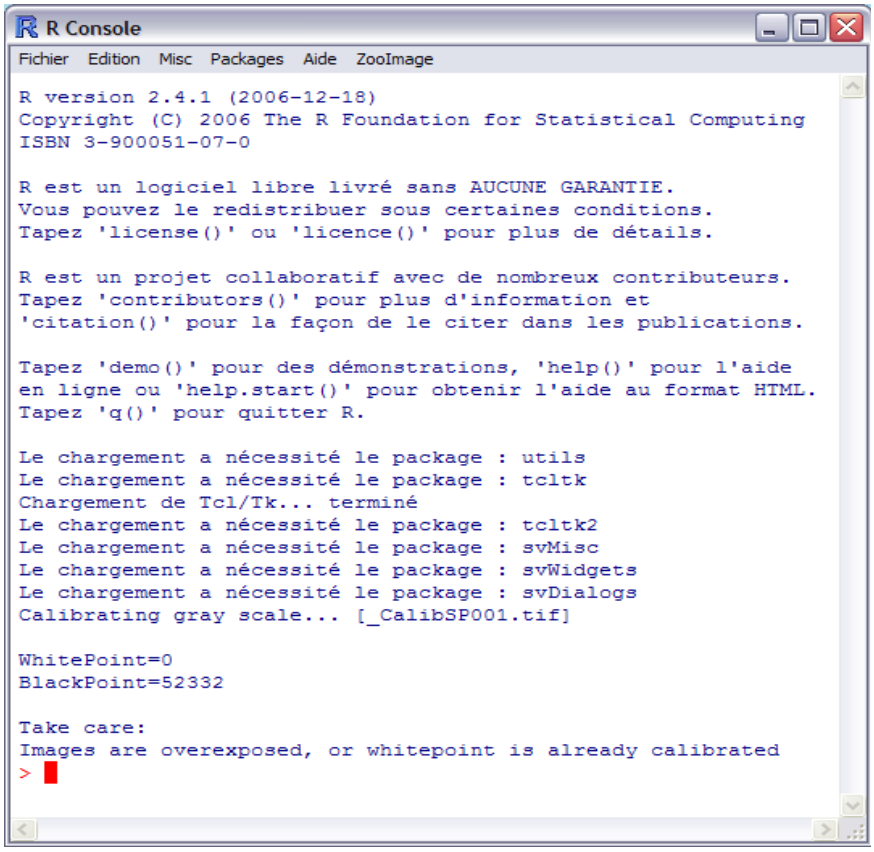
The last part is the inversion of image in order to obtain a black background with white particles.

In Zoo/PhytoImage, the calibration is done automatically using one image containing a part of each grey filters and background. This image has to be saved as tiff format.
Select the Calibrate grayscale (16bit) option in Utilities menu of the Zoo/PhytoImage 1 assistant.



Then you have to select the calibration image and Zoo/PhytoImage does all the process in order to give values of white and black points. They are calculated using mode of grey levels in the image containing grey filters.



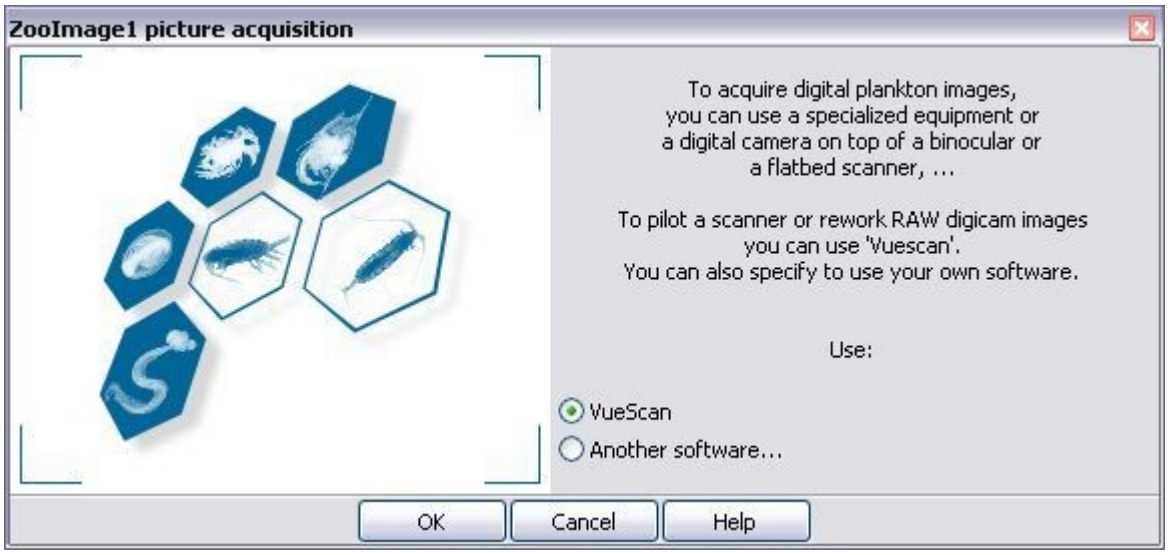


These two parameters will be used for calibration of all images analysed. In the demonstration, a warning message appears because the white point is already calibrated: automatically done with the scanner we used.

1.2. The “acquire image” tool

In this manual, we will use examples downloadable on the software web site. So, you do not need to acquire your own image to practice with Zoo/PhytoImage. In this section, we will show you how you can get your own images using Vuescan.

 To start your image acquisition software from the Zoo/PhytoImage assistant window, use the menu entry Analyze → Acquire images..., the shortcut Ctrl+A, or click on the first button in the toolbar.



This dialog box let you the choice to start Vuescan or any other acquisition software. Select Vuescan and click OK.

Once the software is registered, you can switch to an advanced mode by clicking on the corresponding button at the bottom (if Vuescan is started in “Guide me” mode). You have to parametrize Vuescan according to your acquisition device (digital camera or flatbed scanner) and the images format. Vuescan allows you to save both uncompressed TIFF files with 16-bit grey levels or JPEG 24-bit color files. These two file formats correspond respectively to the “Scanner Gray16” and “Scanner Color” plug-ins in ImageJ (Zoo/PhytoImage plug-ins).

If your acquisition system consists to a digital camera connected to a binocular or a microscope, a simple solution is just to record your images as 16-bits JPEG files, out of any modification by the software. This kind of image can be analysed with the “Microscope Color” Zoo/PhytoImage plug-in (More plug-ins will be developed in the future). Note that, for further use of Zoo/PhytoImage, you are better not to rename these images; they are usually called as Aprefix_xxxx.jpg (where xxxx is an incremented number). Zoo/PhytoImage will allow you to indicate which images correspond to which sample (later on). You should also

record the magnification you used to take the picture, and convert it into a spatial calibration data (size of a pixel in µm). You are supposed to use the same magnification for a series of images (you can split your sample into several fractions and may digitize them at different magnification. Of course, the same resolution has to be used for the same fraction.

Vuescan offers a wide range of options for digitizing your pictures. A couple of options are very sensitive in the context of your image analysis. Additional documents are in preparation to best list Vuescan options for several digitizing devices.

2. Importing images

To start your image importation from the Zoo/PhytoImage assistant window, use the menu entry Analyze → import images, the shortcut CTRL+I, or click on the second button in the toolbar: 

The image importation is indeed performing several tasks to make sure that your pictures are in correct formats and all required metadata are associated. In the current version, you have the choice between three methods of importation:

(1) Just check the presence of metadata files. For this simplest method, you are supposed to rename the images and build the .zim files (Zoo/PhytoImage Metadata) manually. Depending if your pictures are in TIFF or JPEG format, choose the 'Tiff image files (*.tiff)', or 'jpeg image files (*.jpg)' in the same importation dialog box, and select one of the images you want to import (Zoo/PhytoImage will process all images of the same format it founds in a given directory). See "ScanCol24-example" to see an example of this kind of importation.

(2) Compile metadata and rename images according to a specific file called Import_xxxx.zie (zie extension means "Zoo/PhytoImage Import Export"). It is a file that define to Zoo/PhytoImage the way of processing the data step by step. This importation is most adequate for a batch process of large dataset. For this second method, once an Import_xxxx.zie file is built (either with the third importation method, or manually), you can start to import your images by selecting Analyze → Import images..., and choosing the 'Zoo/PhytoImage import extensions (Import_*.zie)' file type in the importation dialog box. Then, you select the Import_xxxx.zie file and let Zoo/PhytoImage does the work! As usual, a log file is displayed at the end of the process and shows explicit information about possible

warnings or errors during the importation process. See the "ScanG16-example" to see an example of this kind of importation.

(3) This method creates automatically an Import_xxxx.zie file using an ImportTemplate.zie (template file) and an Excel/OpenOffice Calc or plain text table. This table gives information about your different samples and the images associated to these samples. The .zie file created after this step is used like for the second importation method. You can see the third method of importation (more advanced) which renames your images and builds automatically their associated metadata files by managing with the "MicroCol24-example".

To correctly realize an importation different precautions have to be taken:

Make sure that all the images you want to process are in one directory on your hard disk. Do not mix pictures you want to process with other ones (with some useless for the analysis) in the same directory.

Keep them separate. For instance, have one *D:\ImageProcess* directory where you store your fresh images and place them in one *D:\ImageDone* directory as soon as they are processed.

Make sure your images are in a correct format: uncompressed TIFF with 16-bit grey levels (preferably with a resolution of 2400dpi) for the Scanner Gray16 plug-in, 24-bit color JPEG (preferably with a resolution of 600dpi and with the lowest compression level) for the *Scanner Color* plug-in, 24bit color JPEG too, but with any resolution for the *Microscope Color* plug-in, …. Other file formats will be accepted in the future. Also, you can use XnView bundled with Zoo/PhytoImage to convert wrong file formats in batch, if needed (see XnView online help).

Make sure you respect the naming convention imposed by Zoo/PhytoImage, which is:

SCS.[YYYY-MM-DD].SS+PP[.II].EXT

With this convention, the images are easily identifiable in a large series, both by the software and by the user. In particular, sorting files alphabetically results in a chronological sorting of the images, according to sampling dates.

SCS is the identifying code of the "Series - Cruise - Station". Use three to four letters to identify the point within all you series/cruises/stations data. Alternatively, you can use any short string that you like to identify your samples.

YYYY-MM-DD is the date of sampling in year-month-day format. If for some reasons the day or the month is unknown, use 00. This is optional.

SS is a code to uniquely identify each sample (useful when several samples are taken the same date at the same station).

PP is the image identifier. Zoo/PhytoImage manages different images per sample, and even, images of different fractions at different dilutions of the same sample. Zoo/PhytoImage will carry all required calculations, including collecting together results from the six images in a single .zid file, calculating abundances and biomasses per m^3, taking into account the two fractions at different dilutions, etc.

II is the sub-image identifier. This is optional and is used in case you have several (sub)-images to cover a single, larger, picture of your sample/fraction.

EXT is the file extension according to the file format. It must be tif (lowercase) for TIFF images and jpg (lowercase) for JPEG pictures.

You do not have to conform to the Zoo/PhytoImage naming convention of the images. However, the minimum is to use NAME+PP.EXT with whatever string you want that uniquely identify one sample, being at least A if you have only one image per sample, and EXT as above. Thus, as a minimum, TIFF images should end with +A.tif and JPEG images with +A.jpg.

For instance, your samples have two fractions (less and more than 1000µm) and different dilutions are applied for the 'large' fraction and the 'small' one. Just decide to call your large fraction 'A' and your small fraction 'B'. Now, if you take three pictures for each fraction, PP will be A1, A2, A3, B1, B2, B3, respectively for the six pictures related to the sample.

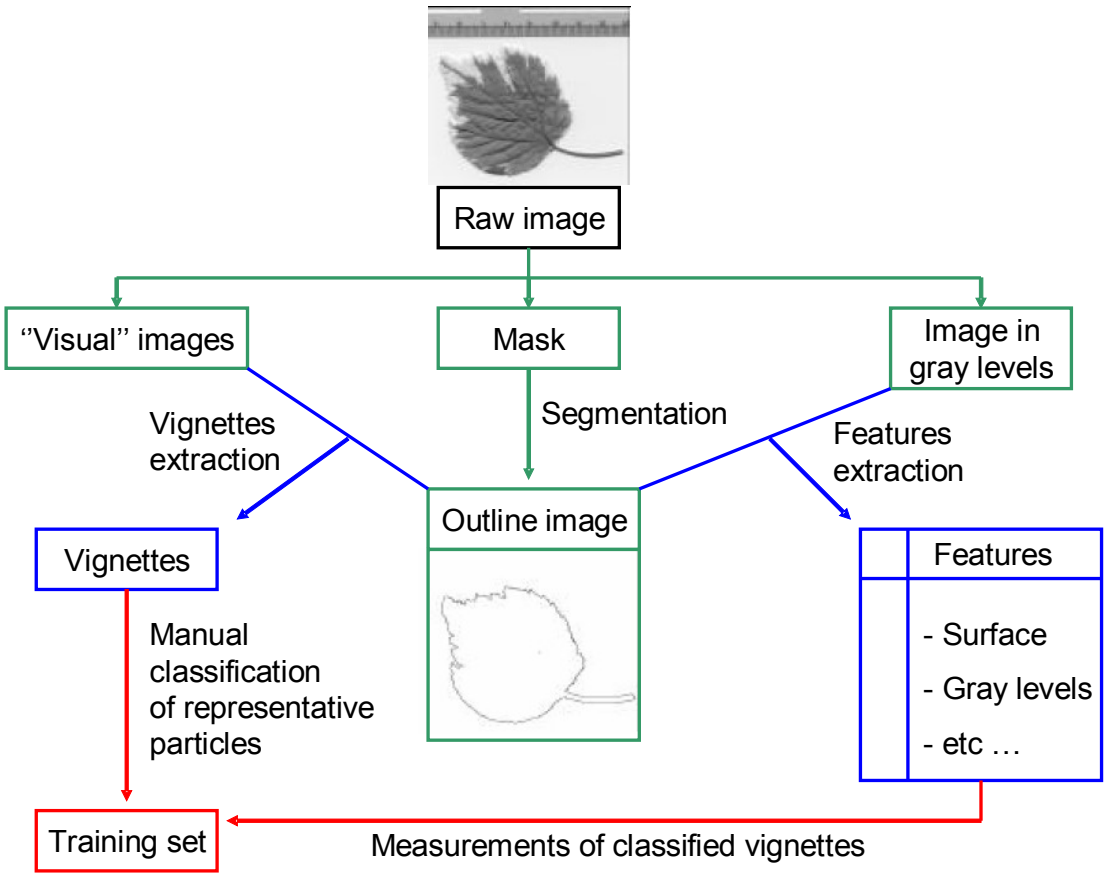
Remind you can automatically rename your images to respect the Zoo/PhytoImage convention during the importation using .zie files. Moreover, for large image data sets like FlowCAM and VPR samples or for long series of microphotography of individual particles, it is obvious you will not fill one zim file for each image. It is easy to rename your images with an automatic method able to work in batch. So, you do not have to rename your images one by one. From a simple table created in Excel or OpenOfficeCalc and an ImportTemplate.zie you can run this importation method. Notice that all the files have to be located in the same directory. Zoo/PhytoImage creates automatically all zim files using a table of parameters and one template.zie for each directory.

In the future, importation and image analysis will be united in one step like for the FlowCAM-example. Like this, in ImageJ, the developers created a specific plug-in to the data obtained by the FlowCAM to import and analyse image in one step.

Do not forget to fill information about white and black points calculated during calibration step.

3. Process images

Once acquired, your images associated to their metadata files are ready to be processed in ImageJ, a free image analysis software. In Zoo/PhytoImage, the general principle of image analysis is always the same. Directly from raw images, the software calculates different images. A "visual enhancement" image designed to enhance details of each particle, a binary image called "mask" designed to differentiate particles of interest from background and an image in grey levels designed to be measured. The mask is used to calculate the "outline" image by segmentation. This image is used to extract features from image in grey levels and to create vignettes (little images of each particle containing a scale).

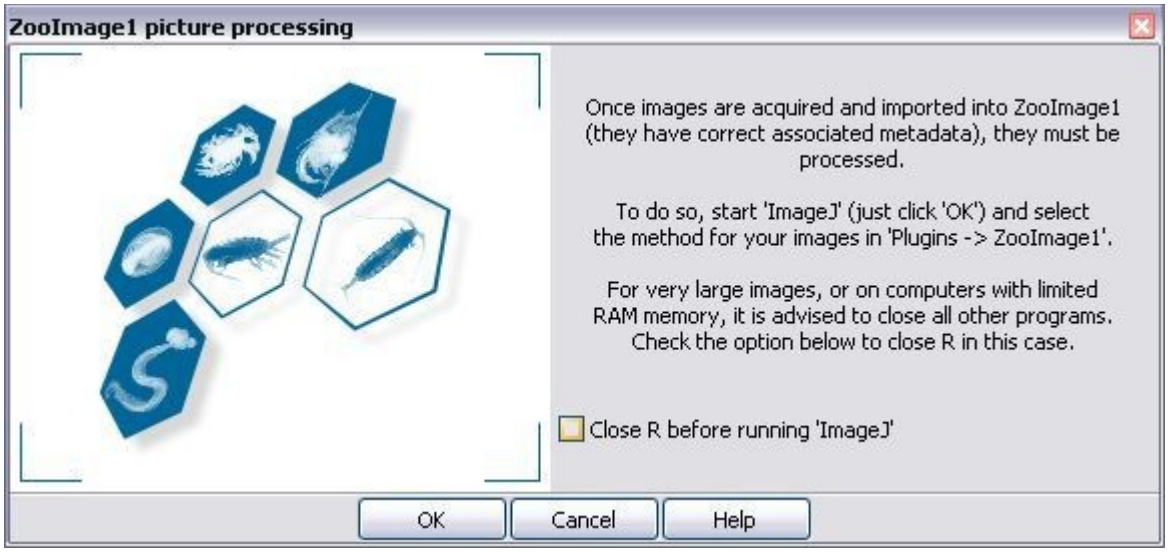


General scheme of image analysis made by Zoo/PhytoImage.

Just before switching from Zoo/PhytoImage to ImageJ, a dialog box proposes you to close Zoo/PhytoImage to do not saturate your RAM memory. There is a minimum RAM memory required by the image process depending also on your images size. The small example images we are dealing with do not require many RAM. So, if you have something like 512Mb on your machine, you should be safe to keep both Zoo/PhytoImage and ImageJ opened simultaneously. If you analyse very large pictures, you should close Zoo/PhytoImage and all other running software **before** starting your image processing in ImageJ. As an example, 16bit grey pictures of 60 million pixels (for instance, 10000×6000 pixels, "ScanG16-example") require 900Mb of RAM allocated to ImageJ. You need at least 1Gb of actual RAM in your computer for dealing with such images.

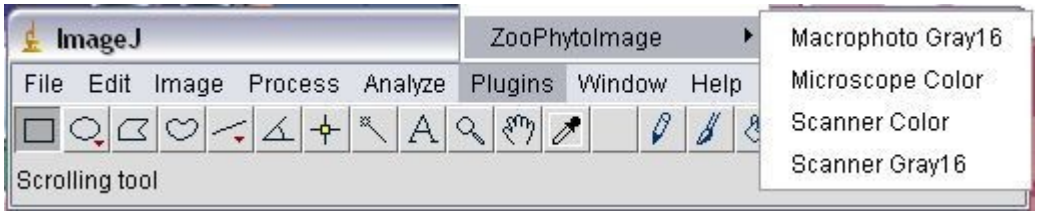
The maximum amount of RAM you can allocate to ImageJ is system dependent. On Windows, do not try to allocate more than 1.6Gb to ImageJ, or the program will crash! Of course, you need at least 2Gb of actual RAM in your machine to use that maximum value. Although we did not tested the Scanner Grey16 *plug-in with images larger than 10.000×6000 pixels, the maximum RAM value should work with images of about 100 million pixels. Thus, currently the largest 16bit grey images you can deal with in ImageJ is something like 10.000×10.000 pixels. At 2400dpi, it is a little bit less than 10x10cm of cell size. If you have larger cell area, just take several separate pictures and both ImageJ and Zoo/PhytoImage will take them into account (you just loose measurement on objects that are cut at the edges of the composite images).*

Start now image processing by click on the third button: 



Click OK in the dialog box and the Zoo/PhytoImage 1 assistant window is minimized and replaced by the equivalent ImageJ main window as:

Zoo/PhytoImage plug-ins are collected together in the menu Plug-ins → ZooPhytoImage.



Plug-ins menu contains also FIT_IMS and VIS plug-ins used to analyse images from FlowCAM and PlanktonJ used to explore zid files, measure parameters on vignettes but also create a manual training set.

In the current version of Zoo/PhytoImage, four plug-ins are available: **Macrophoto Gray16** for analysing images from Madagascar samples digitized with a Macro camera; **Microscope Color** for analyzing phytoplankton micro images from an Ifremer data set and digitalized with a camera placed on the top of a microscope; **Scanner Color** for analyzing color images from SpainBioman data set digitalized under a flatbed scanner; and **Scanner Gray16** for analyzing grey levels images from Madagascar samples digitalized under a flatbed scanner. Whatever the plug-in selected, you can process all your images in batch. A log file gives you information during and at the end of the analysis. This log file allows you to check the advancement of the process and eventually the possible errors. During the process, Zoo/PhytoImage creates in your active directory, various files and subdirectories:

- a "_raw" directory containing raw images compressed in .zip format (or not depending on the options you chose). This directory aims to be stored on external support if image analysis is successful.
- a "_work" directory containing images calculated during the image analysis like mask, grey levels images, "visual enhanced" images and one _dat1.zim by sample analysed. Those images created are useful to know if parameters of image analysis are correct. The _dat1.zim file contains general metadata filled for the importation step but also metadata about processing and all measurements done on each particle. This "_work" directory can be deleted if image analysis is successful.

- one directory by sample analysed. This directory contains all vignettes, their _dat1.zim files associated and one _dat1.Rdata. This last file is a specific R data file that contains the same information as _dat1.zim but is assigned to be process in R and thus easily accessible for the rest of analysis.

The .zim files filled or created during the importation step are located in the process directory and can be stored with raw images if the image analysis is successful. This is automatically done during .zid files creation (see next chapter).

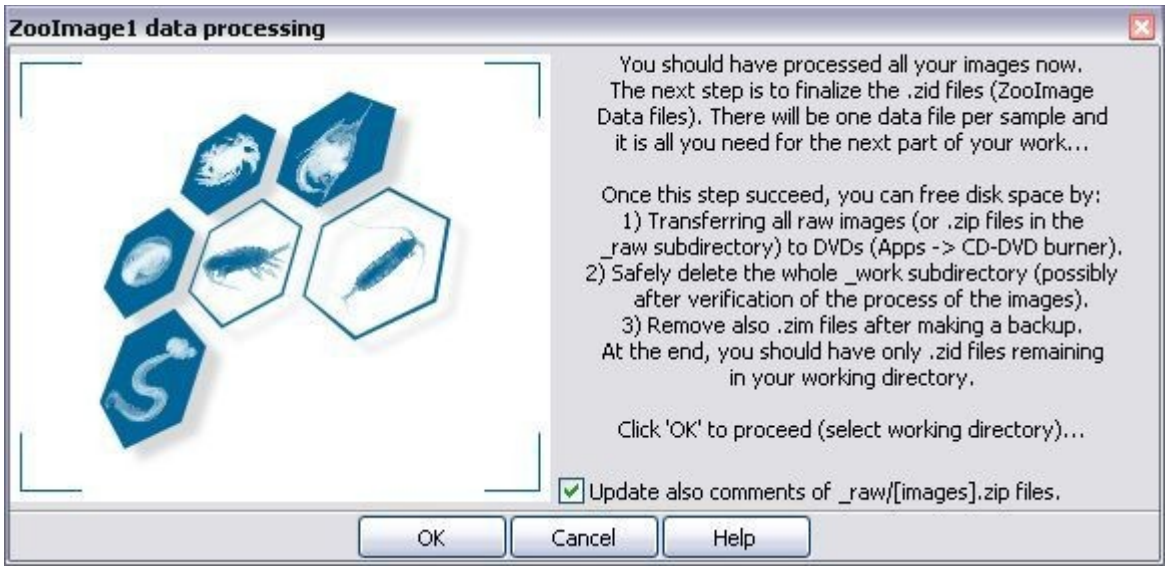
When the image processing is done, you can close ImageJ and return to Zoo/PhytoImage (either restore the Zoo/PhytoImage 1 assistant window, or restart the program).

4. Create .ZID files

The first part of your analysis (import and process of your images) is almost done. You have now to create the **.zid files**. Each zid file is a compressed file of the sample directory created during image analysis. They contain vignettes calculated of a sample, their associated tables of measurements (remind you that a sample can contain a lot of raw images) and only one _dat1.Rdata. This specific *Zoo/PhytoImage Data* files contains all the necessary information for the rest of the analysis and is saved in a zip format to gain disk space as much as possible. These .zid files represent a convenient solution to keep all required data of a long serial sample (thousands of images) on a standard hard disk of 100-300Gb. In such a case, high-resolution raw images easily consume **terabytes** of disk space and cannot be all saved on the current hard disk at the same time! Just process your serial samples bit by bit, and backup raw images from time to time to solve the problem.

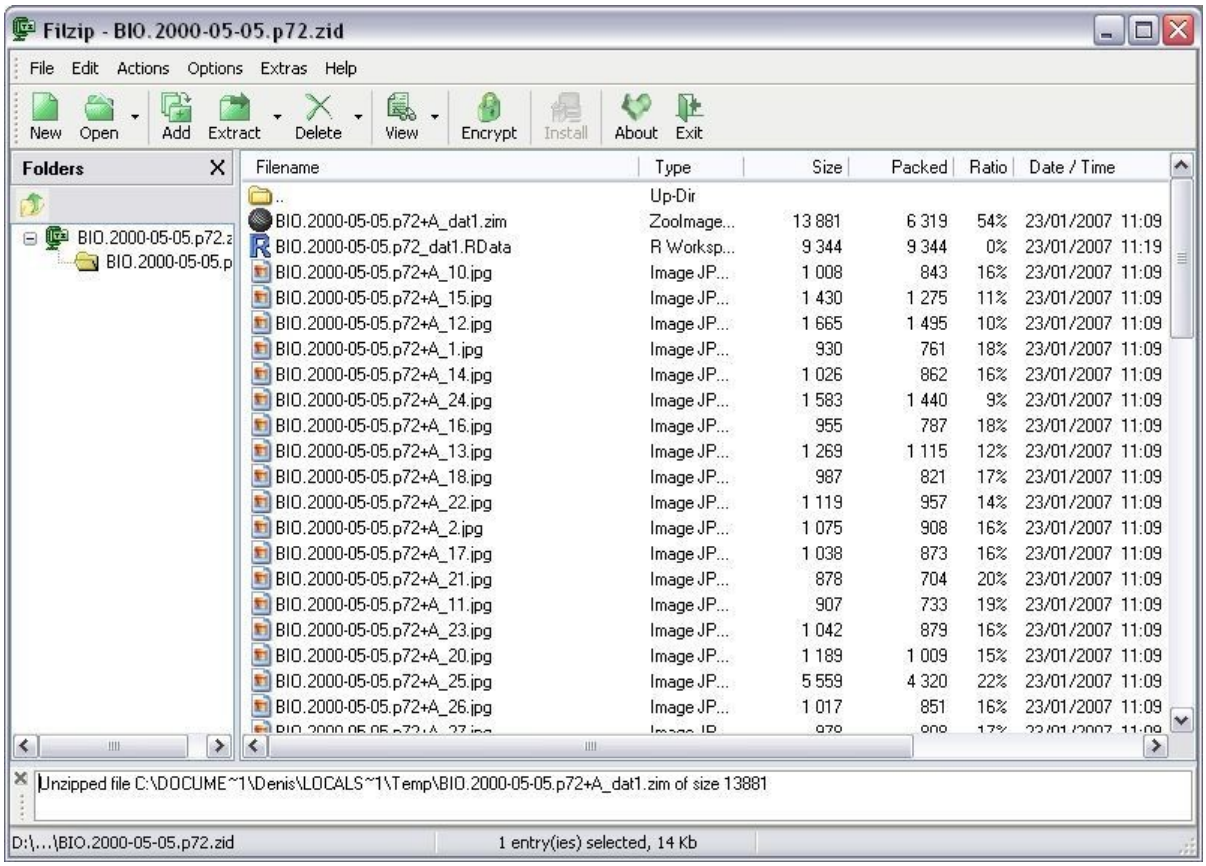
Now, click on the fourth button in the Zoo/PhytoImage 1 assistant: 

The following dialog box appears on the screen:



Instructions should be clear. By clicking OK, you compute .zid files for your processed samples. The option "**update also comments of _raw/[images].zip files"** add .zim data as comments to zipped image files (if you selected that option in the process). [*Since we did not zipped images, we should uncheck that option now and click OK*].

The select the working directory containing subdirectories of samples (one directory by sample).

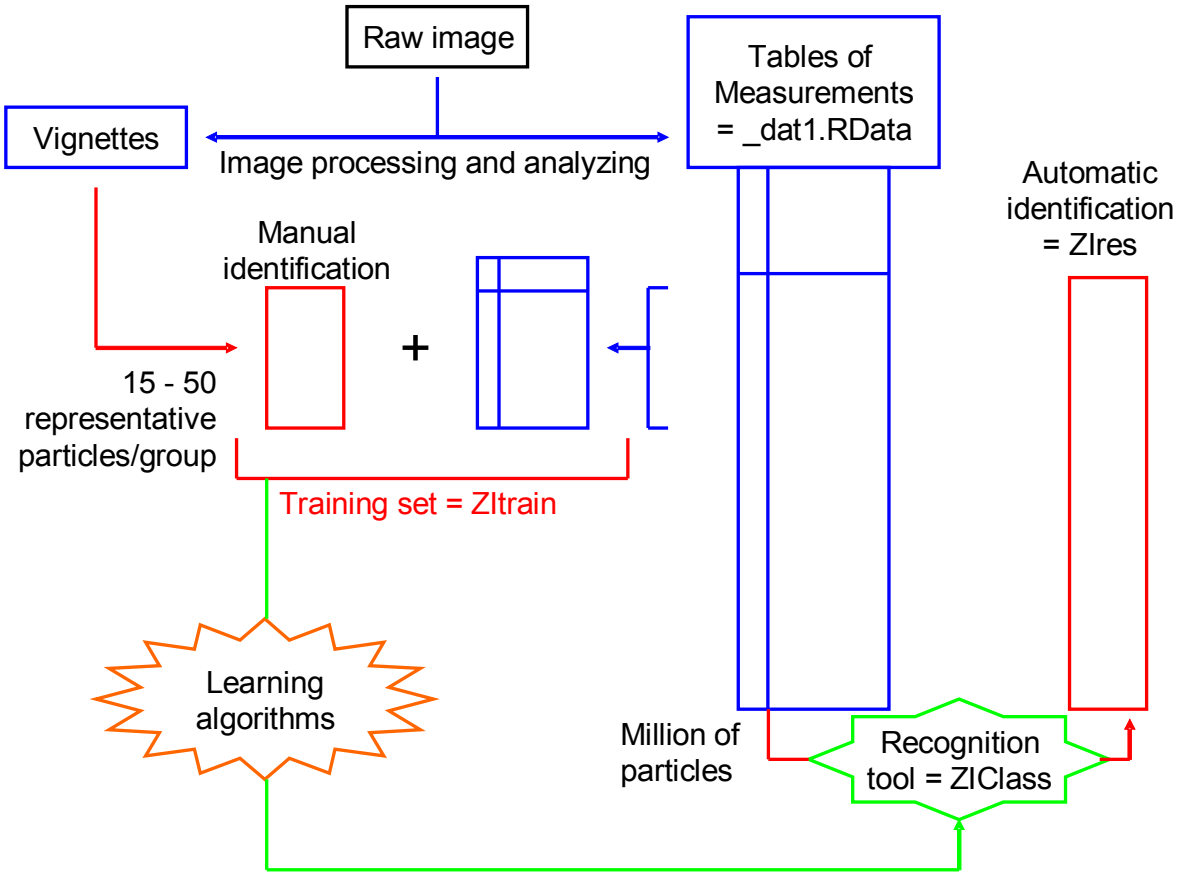


Example of a zid files. As you can see, it contains all the vignettes of your serial sample, the _dat1.zim and the_dat1.RData file.

During zid files creation, the software cleans the current directory in order to free space on hard disk. It deletes "_work" directory and adds zie and zim files to "_raw" directory. Once it is done, you can save _raw directories on external support and only keep on your current computer .zid files that will be used during the rest of analysis.

Section II: Automatic recognition

The first section of Zoo/PhytoImage is now completed, all your images are analysed and only zid files are still remaining on your hard disk. This second section focuses on the creation of an automatic recognition tool and on the test of its performances. This step uses methods of machine learning, it consists to teach the computer to discriminate particles in different groups. In this sense, you first have to create a manual classification of vignettes in different groups, called training set. This training set will be used by machine learning algorithms in order to calculate a recognition tool. Zoo/PhytoImage allows you to create a personal training set and studis it with 6 different and widespread machine learning algorithms.



General scheme of image processing and automatic recognition. Representative vignettes of series variability are manually classified in different groups on hard disk. These images coupled with their measurements constitute the "training set". Machine learning algorithms read this manual classification and create a recognition tool. This tool can predict the group of an object using only their measurements.

In Zoo/PhytoImage, different R objects are created at each important step: the training set, the recognition tool and the automatic recognition (more exactly results) are named ZItrain, ZIclass and ZIres respectively. All these objects can be saved on hard disk as R data files.

1. Manually classifying vignettes

The training set is a manual classification of vignettes in different taxonomical groups and sub groups. These vignettes are representative of the variability of your samples.

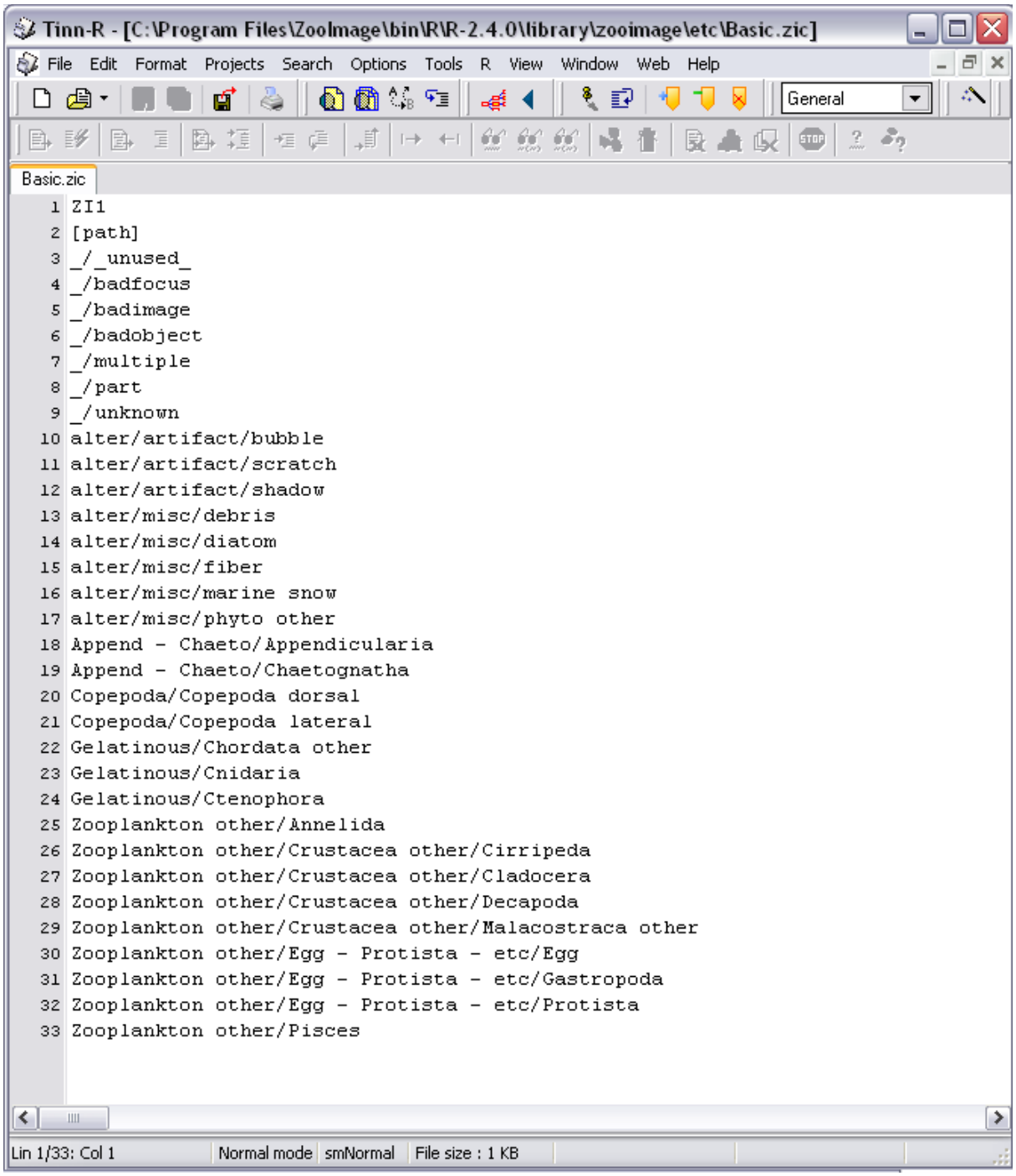
You first have to decide which samples (zid files) you will use for the creation of the training set. Select some samples (i.e., a couple of .zid files) that are representative of the

whole variability in your temporal series. Select samples that span on the whole time scale (possibly several years) and the whole considered geographic area. Select also samples collected at different seasons, if this applies. Knowing the average number of vignettes you have in a sample, you can determine how many samples you need. A complete training set contains usually between 2000 and 3000 vignettes; a couple of ten zid files should be enough, but it depends of the number of groups you want to recognize.

If you want to carry out a rapid (or not) training set, starting with a long historical serial sample already available in your laboratory, it could be interesting to first select images of representative samples that will be include in the training set and digitize them in priority. That way you do not have to wait the digitalization and the processing of all the samples in the series to make your training set! Also, if different people are digitizing the sample (technicians) and making the training set (specialized taxonomists and biostatisticians), you could work in parallel once the few samples required for the training set are digitized.

Once you have chosen the representative zid files (in a Zoo/PhytoImage window), you have to create your training set. The software proposes you three defaults training set hierarchy but it is obvious you can create your own one. In this sense, you have to create a text file with a zic extension (Zoo/PhytoImage Classification) that contains your own hierarchy for the classification.

The three defaults zic files are located in the bin\R\R-2.4.1\library\zooimage\etc subdirectory of your installation directory.



This figure shows you the structure of the basic training set available with the software. Subdirectories are separated by "/" and a "_" directory has to be created.

To create your zic file you have to respect two obligations. You first have to write the first two lines (ZI1 and [path]) because they allow the software to create your classification on hard disk. You secondly have to create a "_" directory. This directory can be assimilated as a "garbage" group. It should contain all vignettes you will not use for the training set realisation but also groups containing not enough vignettes to be recognised.

Make sure you use **unique names** for **all levels** of all groups. Do not use a classification like *Nauplius* subgroup in *Copepoda* and *Nauplius* subgroup in *Malacostraca*. Indeed, the program will manipulate groups independently for some treatments and how to differ *Nauplius* from *Nauplius* then, when you don't use the grouping hierarchy? Correct classification should be: *Copepoda nauplius* in *Copepoda* versus *Malacostraca nauplius* in *Malacostraca*.

Remember the special organisation of Zoo/PhytoImage training set with its hierarchical structure of directories respecting filiations of groups. Indeed, you can have a relatively complex organization of the different particles (taxa, ecological groups, or any other grouping of the plankton that suits your needs) in a **hierarchical tree** where leaves are groups. Hence, you have relationship between the groups (for instance, *Sapphirina intestinata* and *Sapphirina ovatolanceolata* are collected together in the *Sapphirina sp* group. *Copilia sp* and *Sapphirina sp* form your *Sapphirinidae* group. *Sapphirinidae* together with *Oncaeidae* and *Corycaeidae* (which contain also corresponding subgroups) are collected together in the *Poecilostomatoida*, etc. Up to the top group called *Copepoda*.

You can also decide to create other groupings, like ecological groups, or even mix the styles. You are here 100% free of the groups you create, but there are a couple of constraints: (1) make logical hierarchy of your groups and subgroups; (2) keep in mind the parameters (abundances, biomasses and partial size spectra) that you want to calculate on these groups; (3) make only groups where you can actually classify vignettes with a reasonable accuracy solely on the visual inspection of these vignettes; (4) it is useless to make groups for very rare items –you need at least fifteen to twenty representative vignettes in each group in your training set, 30 to 50 is even better–; (5) Finally, the most pertinent grouping is the one that can be discriminate by the computer with a reasonable accuracy!

To fill in groups, you have to manually classify all kinds of items, even those you are not interested in (may be, bubbles, marine snow, phytoplankton if you are only interested by zooplankton, etc...). Indeed, all kinds of particles representative of one group need to be assigned in a group to avoid misclassification. You have to recognize those items in order to eliminate them from the counting... and you need a group in the training set for that!

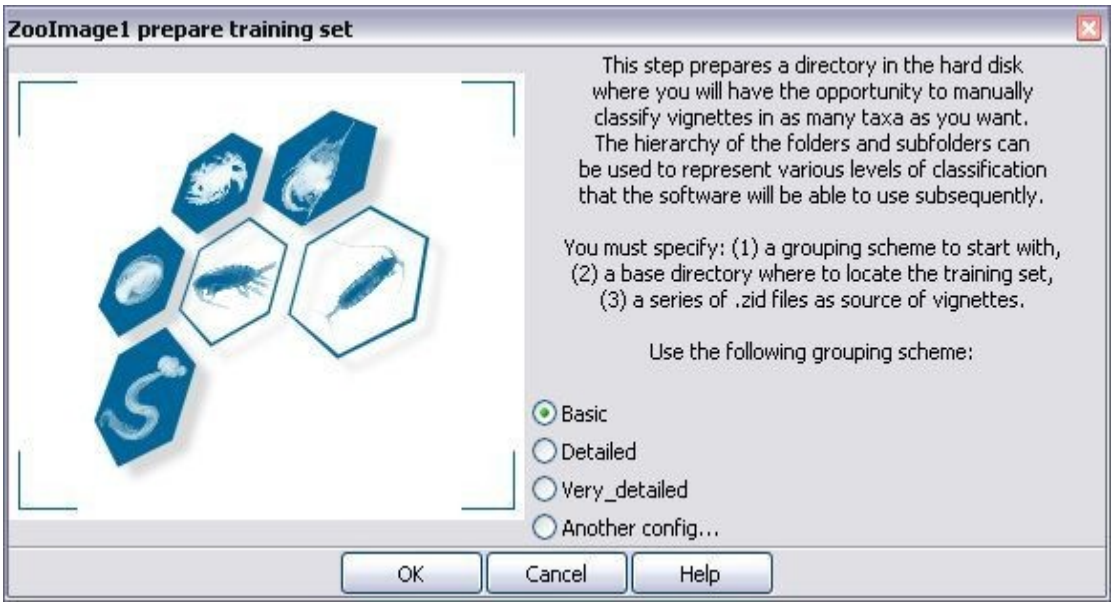
You don't need to classify **all** vignettes of a same group. When you have between 20 and 50 items in one group and you think it is well representative of the overall variability of a

same group, you don't need to add more vignettes. Also, fuzzy objects, unrecognisable ones, multiple or part (except for VPR images), rare taxa, etc. do not need to be classified. Aberrant individuals which are not likely to occur often in your samples should be eliminated too. You have a special top group named '_' in the hierarchy for all these items. If, at the end of the classification, you do not have enough vignettes for one group in your training set, it is better to place them in the "_" directory but do not delete them. **All vignettes in the '_' top group or any of its subgroups will not be considered in the training set.**

For biomasses calculations, it could be useful to further split groups depending on the orientation of the animals: conversion formulas could be different for 'lateral' or 'dorso-ventral' views of the same animals. (Create subgroups for them, if you want to take advantage of these different conversion formulas). Ex: *Oithona sp. lateral* versus *Oithona sp. dorsal*.

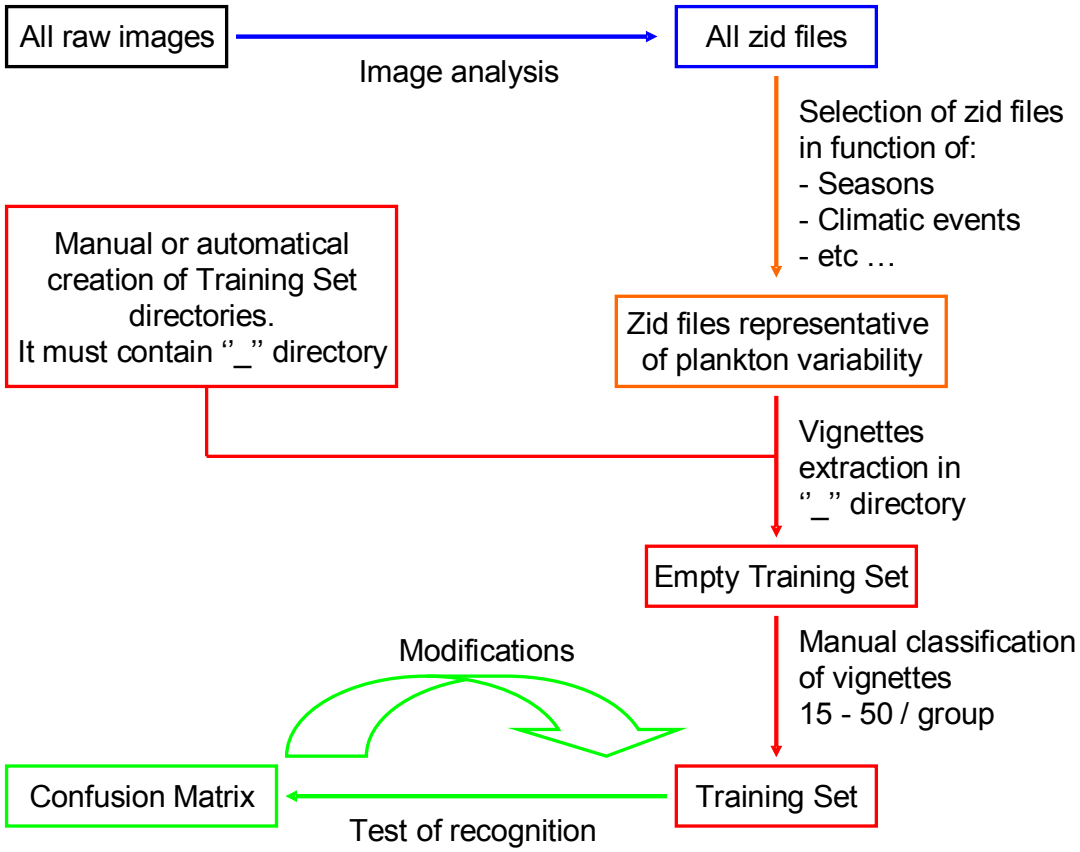
 Now, click on the fifth button on the Zoo/PhytoImage 1 assistant toolbar:

A dialog box with instructions appears on the screen that allows you to decide the taxonomical levels of the default training set. You can also select the zic file you created (Another config...) but remind that you have to create a "_" directory where vignettes will be copy from zid files. At this step, you have to select representative zid files of your serial samples. Then, the software creates your classification on hard disk and extracts vignettes to the "_" directory.



The software extracts vignettes from zid files selected to the "_" directory and opens XnView. Then, you have to fill each group with representative vignettes to finally obtain between 20 and 50 vignettes per class.

The training set is manually realized in XnView (http://www.xnview.com). XnView is a free software for non commercial use. It is both an image viewer/manager and an image converter. Here, we only use its ability to work with thumbnails of images in directories and manage them. We don't use all its features!



General scheme of training set realisation in Zoo/PhytoImage.

You can modify your training set to optimise the accuracy of your recognition tool by analysing the confusion matrix (see below). If you want to further complete your training set with other zid files, do not forget to extract the corresponding _dat1.RData in the basal root of the training set. Indeed, the learning phase uses measurements of _dat1.RData to create the recognition tool.

Making a functional manual training set is difficult and time-consuming. We still not have all the required knowledge to best define groups and strategies in this field. Also, it is a sensitive step of the whole process where best cooperation between taxonomists and biostatisticians has to be seeking... and these two specialization share very little common words in their languages. So, it complicates the task even more! You should consider yourself

as a pioneer here! **Share your experience!** The Zoo/PhytoImage forum community is waiting for you.

2. Reading a manual training set from disk

Once you are satisfied with your manual training set (or after reworking it, guided by the inspection of the confusion matrix, see hereunder), you have to read it in Zoo/PhytoImage.

Click on the sixth button on the Zoo/PhytoImage1 assistant toolbar: 

When you selected your training set, the software reads your manual classification automatically and creates a ZITrain object.

This R object contains the names of groups and the number of particles in each one. It will be later read by one or different machine learning algorithms. The ZITrain is supposed to be reused further in the learning phase. That's why you can save it on hard disk and load it again later (see section 4).

3. Making and analyzing an automatic classifier

In Zoo/PhytoImage, classifier algorithms used are ranged in a category called "machine learning". Basically, you train the algorithm with classification (manual identification of particle) and measurements made on the same objects, to recognize the particles according to their measurements. It is a very simple scheme, but it has proved his efficiency in many situations.

Many algorithms exist, and many are implemented in R over which Zoo/PhytoImage is running. The Zoo/PhytoImage dialog box gives access only to six of them. Moreover, in order to simplify the process, only default values are given for parameters. The solution you will obtain is, thus, often suboptimal.

Zoo/PhytoImage provides a wide range of widespread machine learning algorithms. Linear discriminant analysis is a simple and rapid method to have a quick estimation of recognition tool efficiency. This method looks for the best discrimination between groups by calculation of linear combinations of measurements. Classification trees based algorithms like recursive partitioning tree and random forest. These methods create one or several decisional random trees for which each node discriminates variables in two groups. Principally, random forest is supposed to be the most efficient method but little is known in the literature about the use of this algorithm. Artificial neural network creates a neural web with one or more intermediary layers between input layer constituted by measurements and output layer constituted by groups of the training set. This method proved his efficiency by giving analogue results

between manual and artificial recognition on phytoplankton groups. K-nearest neighbour (knn) and learning vector quantization (lvq) methods determine groups of object to be recognised by minimising distances between them and training set particles. Knn determines groups using K distances between training set objects whereas lvq uses k distances between codebooks: kind of training set summary created by combination of variables. These methods have been successfully used in the automatic recognition of plankton.

More machine learning algorithms can be implemented in a future version of Zoo/PhytoImage like Support Vector Machine, a method currently used in automatic recognition of plankton.

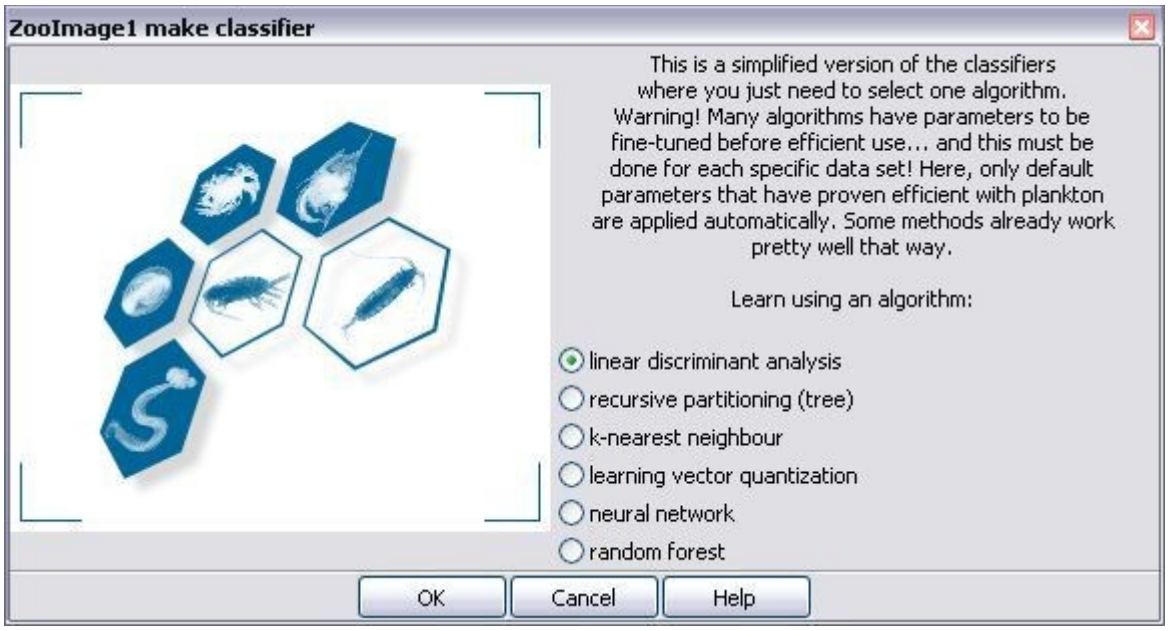
Many "machine learning" algorithms should be put in the "do not try this at home!" category. It means that you need a trained (well-experienced) biostatistician to get the best from them and to analyze results to make sure they produce ***consistent****,* ***reliable*** *and* ***accurate*** *identification of your plankton items. Everything was voluntary simplified in the Zoo/PhytoImage dialog box, just to give you an idea of these algorithms, and to allow you to have a round-trip process of your data in an easy way.* ***Don't be fooled by the apparent simplicity of the process using Zoo/PhytoImage dialog boxes!*** *For serious analyses, consider to fine-tune your classifier with a biostatistician that will use all the functions provided by R (he will program code in R's native language, instead of just clicking with the mouse on a few options in the dialog box). There is no warranty on the results, and we would not endorse responsibility of the consequences for false results published after using "uncertified" 'toys' classifiers!*

3.1. Training a classifier

Having a 'ZITrain' object in memory, you can now create a 'ZIClass' object. This object is created by the learning phase of the algorithm. It can recognize the groups your plankton particles only by using measurements made on them.

Click on the seventh button on the Zoo/PhytoImage1 assistant toolbar: 

A dialog box appears. It displays a warning message about the simplified learning phase and proposes a variety of "machine learning" algorithms to use. During the learning phase, only groups filled manually are used, the empty ones are useless.



Zoo/PhytoImage dialog box presenting the 6 default machine learning algorithms provided with the current version of Zoo/PhytoImage.

When the learning phase is done, its performance is assessed using a method called "10-fold cross-validation". Then, total accuracy and error by group is reported to the R Console.

In order to obtain an estimation of error rate without bias, we have to use different particles for teaching and test algorithm. In this sense, cross validation methods are usually used. The principle is quite easy: the training set is randomly divided in fractions of the same size (for example 10 fractions). The learning phase is made on 9/10 of this training set and the recognition on the last tenth part, called test set. This operation is executed 10 times with, each time, a different fraction used for the test, this one is never used in the training set. We obtain 10 different recognition tools. These tools are used to predict particles of the test set. The automatic identifications are pooled and compared to the manual identification of same individuals using a confusion matrix. With cross validation, each particle is used for the learning and the test phase but never in the same time. The cross validation allows you to measure a global error rate by minimizing bias and by optimizing available training set.

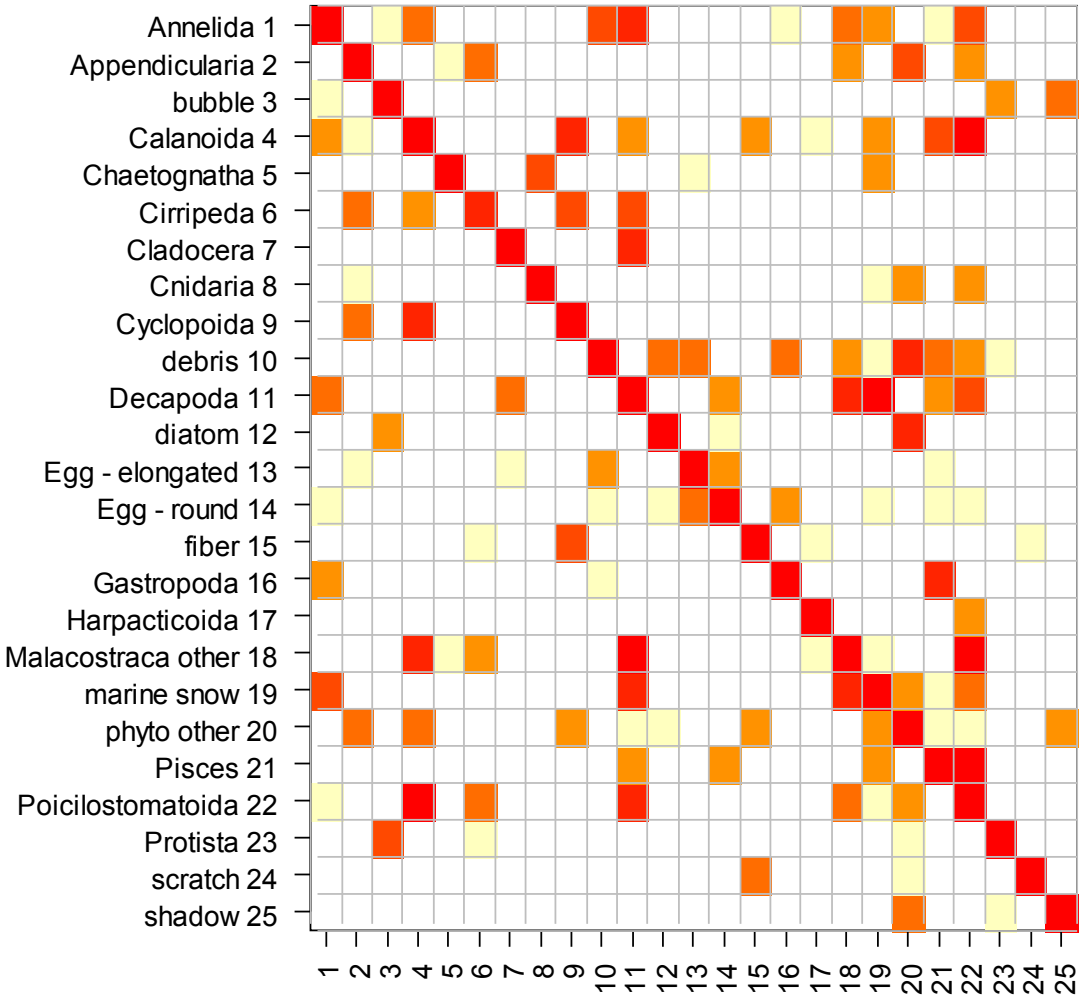
The ZIClass object is a tool planed to be further reused during samples calculation (see part III). You can save it on your hard disk and load it again later (see section 4).

3.2. Analyzing classifier performances

Having a 'ZIClass' object in memory, you should calculate a **10-fold cross-validated confusion matrix** between your manual and the automatic classification. The confusion

matrix is the diagnostic tool of the classifier efficiency. A confusion matrix is a square contingency table that compares all groups of the manual classification with all groups of the automatic recognition. The number of items in each cell corresponds to the counting of objects. The diagonal (from top-left to bottom-right) corresponds to the cells where both identifications are the same. In other words, this diagonal represents the **correct** counting of predicted items. All cells outside of the diagonal depict disagreement in both classifications. They are usually errors done by the automatic classifier, assuming that there is **no error** in the manual training set. To calculate and display the confusion matrix for your classifier, click on the eight button on the Zoo/PhytoImage 1 assistant toolbar: 

The software generates automatically two representations of the same confusion matrix using ZIClass object in memory: a numerical one which appears in R console window and a graphic one which appears in a new R widow.



Graphic representation of the confusion matrix based on 10 cross fold validation. All particles outside the diagonal are errors made by the automatic recognition tool.

4. Manipulating Zoo/PhytoImage objects

You do not have to read manual training sets and train classifiers each time you launch Zoo/PhytoImage. You can **save** and **restore** existing objects. The Objects menu provides functions to do so:

Objects → Load Reloads one or several objects from a .RData file. The .RData file is a binary format used by R to save variables. You can save several objects in the same file and reload them all at once in this case. The .RData files can be exchanged between computers even on different platforms (.RData files generated on Windows are totally compatible with those made on Linux/Unix or Mac OS X).

Objects → Save This option gives you the opportunity to select one or more 'ZIxxx' objects (Zoo/PhytoImage specific objects) installed in the R memory, and to save them in a file.

Objects → List It prints the list of all Zoo/PhytoImage objects currently in memory.

Objects → Remove It permanently deletes one or several objects from memory. Consider using this command to free up memory if you created a lot of objects that you don't need any more.

The .RData files are very convenient to exchange training sets and thoroughly-tested classifiers with your colleagues. Everything is included in the .RData files to reuse those manual training sets and/or these classifiers on a different computer. Note, however, that the function to rebuild a manual classification tree on disk, based on a 'ZITrain' object is not implemented yet in the current version of Zoo/PhytoImage. It is not possible yet to edit a manual training set from its 'ZITrain' object. So for the moment, you can zip the whole tree of the manual training set and send it to your colleagues as well.

R has a mechanism to save and restore automatically all objects in memory when you quit the program and restart it from the same active directory. When you quit R (File → Exit on the R Console, or click the close button of the R Console), you have a question: "Save workspace image?" that appears. If you click No, you quit R without saving. If you click Yes, it saves the data as .RData file in the current active directory (the one reported in the status bar of the Zoo/PhytoImage1 assistant window). It also saves the history of commands in a

.Rhistory file in the same directory. The next time you start R, you can restore this .RData file if you like. **It is far better to use the Objects menu and selectively save/restore given objects than to systematically rely on this mechanism!** This way, you can also choose a meaningful name and directory where you store your data! So, if you save your objects using the Objects menu of Zoo/PhytoImage, you can systematically answer No to "Save workspace image?" when you quit R/Zoo/PhytoImage.

Section III: Calculating, visualizing and exporting series

This section assumes that you have already made all .zid files from your raw images (part I) and that you have a valid and optimised 'ZIClass' object in memory (part II) either that you just created, or that your reloaded from a .RData file. You can now analyse your complete series using these two steps.

The first part discusses how to document your serial sample. A general file containing all information and additional metadata about your samples has to be created (remind a sample is compressed in one zid file). Once done, you can analyse your series using your ZIclass object to automatically recognise all particles in zid files. Next, you will extract biological and ecological information from your samples. This third part of the analysis deals with the calculation of biologically meaningful statistics that summarize each sample: abundances, biomasses and size spectra (total or per taxa). These results are stored in 'ZIRes' objects (Zoo/PhytoImage Results). They are most suitable for spatial and temporal analysis at level series, which can be done in R-Zoo/PhytoImage. If you do not use R for your statistical analysis and graphs creation, you can export tables to analyze them in other software like Matlab, for instance.

1. Creating series

Up to now, all treatments were made at the sample level (zid file). You never had more than one sample loaded in memory. Notice that a sequence of samples is always processed one by one by Zoo/PhytoImage reporting long processes in a log file. So, you can let the software working and come back later to see the results. Zoo/PhytoImage is not designed as a toy program that would be just able to calculate a couple of demo examples, but that will crash with an "out-of-memory" message with any serious dataset!

When we speak about serious datasets in the field of plankton image analysis, it really means **Terabytes of raw images** to process. Since you can backup your raw images and Zoo/PhytoImage cares about **storing highly compressed data in .zid files**, you can really

process very large series containing thousands, or even billions (tens of thousands of samples with a simple PC). You can store, indeed, all zid files in a single hard disk of 200-300Gb.

Zoo/PhytoImage will perform all the calculations: averaging replicates, adding data from the fractions after applying corrections for different dilutions, and rescaling results to express them automatically per square meter unit of seawater. The software uses the following general equation to calculate results in cubic meter of seawater from images:

$$P\ (unit/m^3) = [\Sigma\ P.img\ (unit/image) / R] / F / Dil / Vol$$

Where P is one of the two parameters searched (abundance or biomass),

P.img is the same parameter calculated on each image of each fraction (if sample is fractioned for digitisation),

R is the number of replicates of each fraction ("Replicates" field in zim files),

F is the part of digitalised particles ("CellPart" field in zim files),

Dil is the dilution used for dilution of samples ("SubPart" field in zim files) and,

Vol is the sea water volume sampled in m³ ("VolIni" field in zim files).

There is almost unlimited number of objects in each sample (the current limit is probably around a few hundreds of thousands items per sample, that is, the size of a matrix R can store in memory at once with a 2-4Gb RAM computer). This is not really a limitation because a few thousands to a few tens of thousands of objects are enough to evaluate the composition of a single sample, even for relatively rare taxa (with 10.000 objects measured in a sample, even rare taxa representing 1% of the sample composition will be represented by about 100 individuals).

Of course, processing time is proportional to the size of the series, but Zoo/PhytoImage proposes various mechanisms to recover what was wrong treated after a sample processing, and the error is reported in the log file. So, it is possible to check the error and to reprocess only the problematic sample(s) later on.

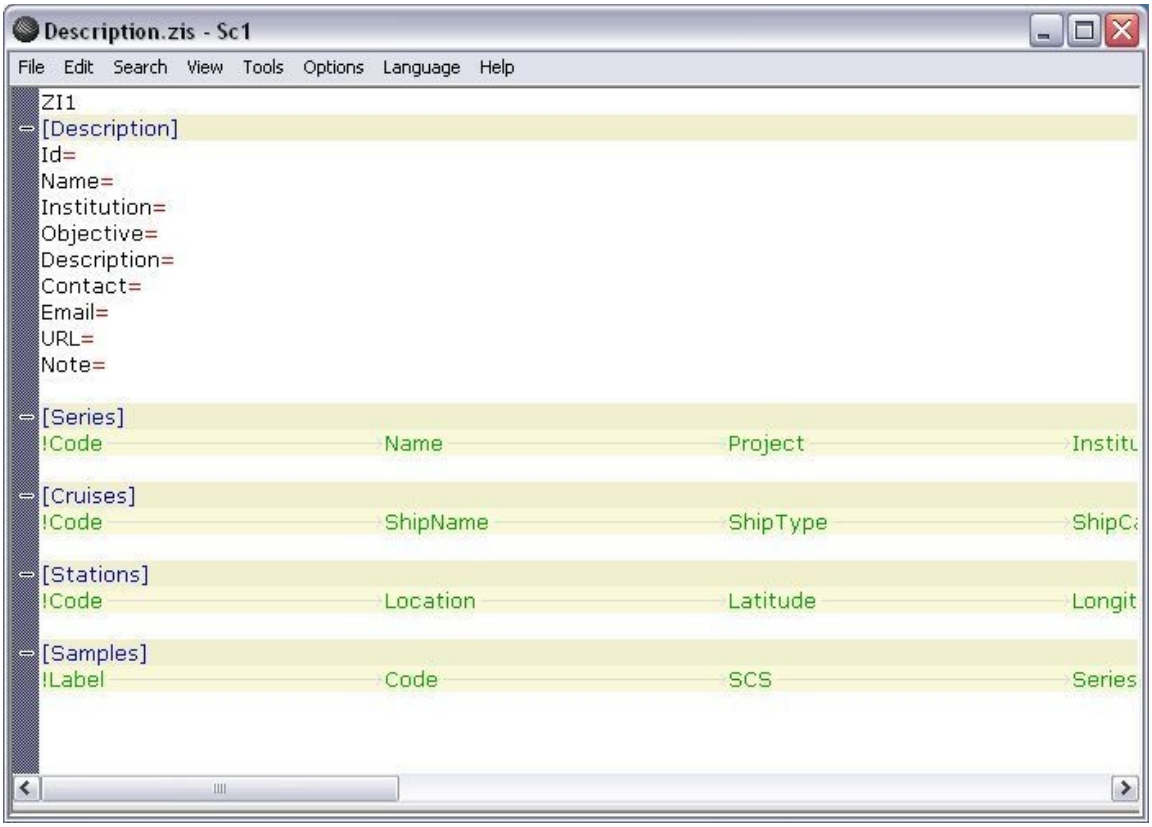
1. Creating and documenting a series

Until now, your .zid files had independent lives, totally ignoring them each other. It is now time to tell to Zoo/PhytoImage which .zid files you want to collect together in a space-time series. This is done by editing a samples description file with a ".zis" extension (Zoo/PhytoImage Series). These .zis files contain all samples you want to analyse and metadata associated like date, temperature, etc. You can create as many .zis files as you need, making thus different series (for instance, a variation in time at a single station for one series,

a spatial coverage of the area at a given time for another series, etc.). The .zis files need to contain all your further analysis of time series, so fill them with a lot of precautions.

Click on the ninth button on the Zoo/PhytoImage1 assistant toolbar: 

The software opens an empty .zis table in which each line corresponds to one sample.



Instructions to correctly fill in this important file are explained in appendix, but you can also have a general idea when you look at zis files available with examples. All fields do not have to be filled in, but try to fill in as many fields you can because they give relevant information that will be used in the rest of analysis.

Zis files have to be stored in the same directory than zid files because they are used to read information from them during calculation of temporal series.

1.1. Biomass calculation

Zoo/PhytoImage calculates biomasses in a non destructive way by using a relationship found in the literature between size of organisms and their biomass. Because this relationship is usually obtained from manual measurement under microscope or binocular, we need to use an additional equation. This second relationship converts one parameter automatically measured

on each particle during image analysis (for example Equivalent Circular Diameter) into the manual measurement provided in literature (i.e length or body area…). This last relationship is obtained by calculating a linear regression between ECD and manual measurements made on vignettes from training set using ImageJ (you can also use PlanktonJ).

The allometric relation between biomass and ECD is expressed below:

$y = (P1\ x + P2)^{P3}$

where y is the biomass (carbon content, dry mass, etc),

x is the parameter measured during image analysis (ECD or other)

and P1-P2-P3 are the allometric parameters.

In the current version of Zoo/PhytoImage, a "Conversion.txt" file has to be created. This file contains a table that opposes groups to their allometric parameters calculated like explained.

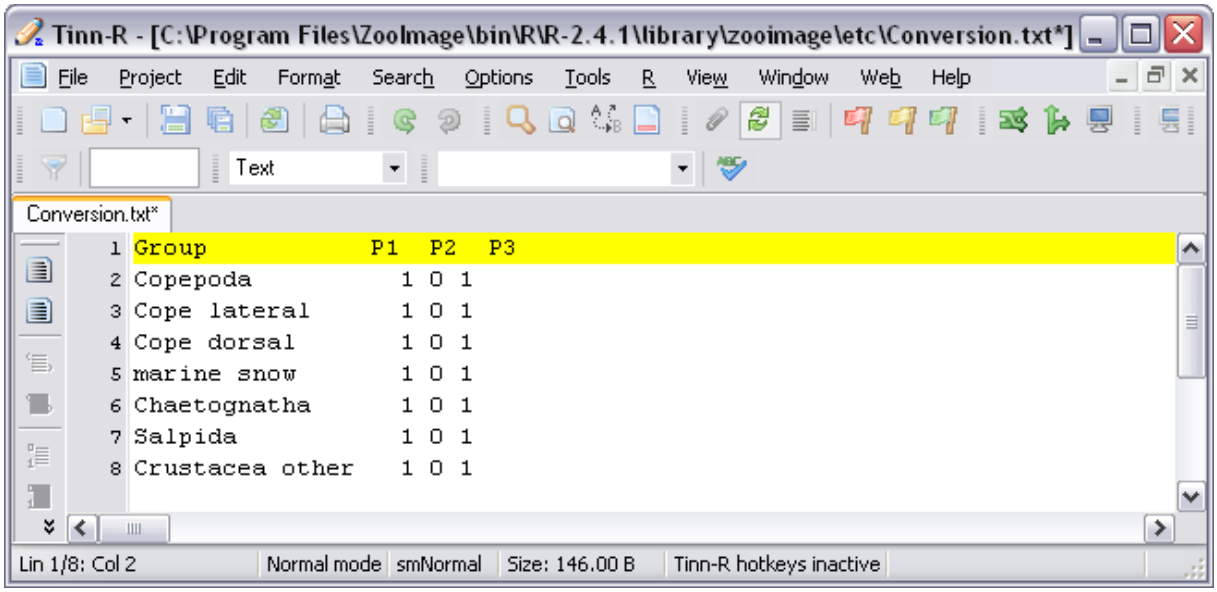


Figure of the "conversion.txt" table available with the current version of the software.

The default allometric parameters of P1, P2 and P3 are 1, 0 and 1 respectively. This general equation is thus limited to the relationship between the biomass and the ECD.

This file has to be stored in the bin\R\R-2.4.1\library\zooimage\etc subdirectory of your installation directory.

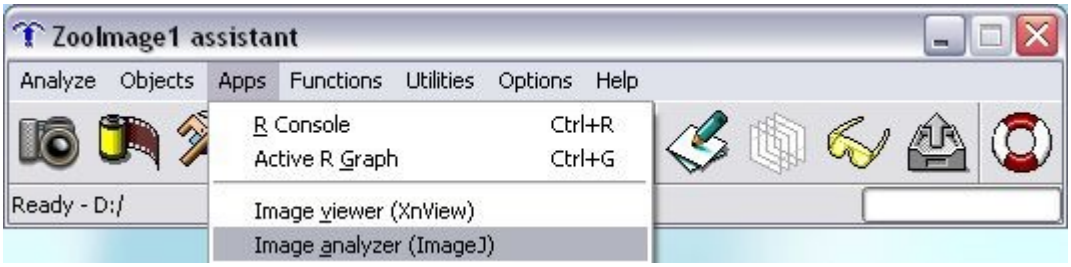
When you have a relationship between biomass and one parameter you can create your conversion table. For example, Nakata *et al.* 2001 determined for *calanoida sp.*:

$C = 4.27\ PL^{2.15}$ ($r = 0.931$, $p < 0.001$)

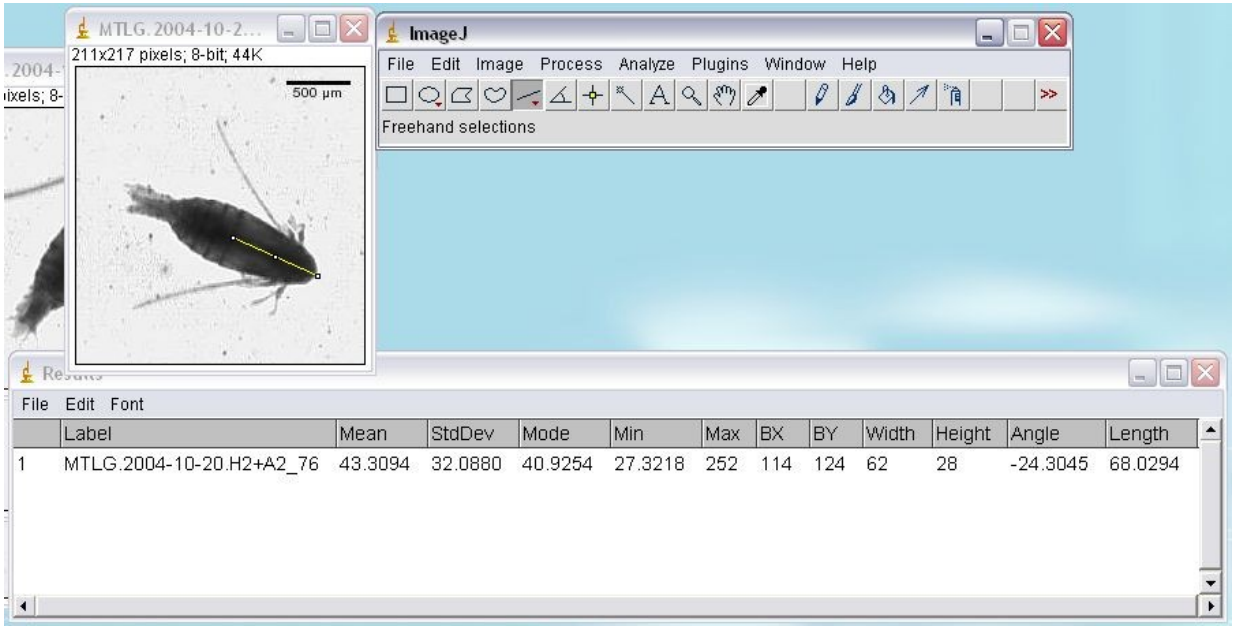
where C is the carbon content in µg C / particle and PL is the length of prosome in mm.

Nakata K., Koyama S. and Matsukawa Y., 2001. Interannual variation in spring biomass and gut content composition of copepods in the Kuroshio current, 1971-89. *Fish. Oceanogr*, 10(4):329-341.

To determine allometric parameters of a training set group, you first you have to manually measure in ImageJ the same parameter than in literature (i.e. PL) on particles from the training set (*calanoida sp.*). These parameters can be measured in PlanktonJ an interface implemented in ImageJ which allows you to visualize particles from zid files or from a training set.

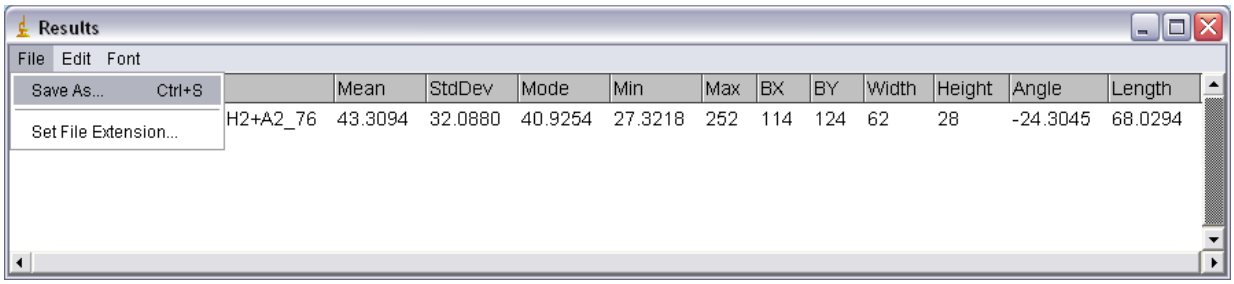


You then have to measure in ImageJ all particles of a training set group, for example "calanoida".

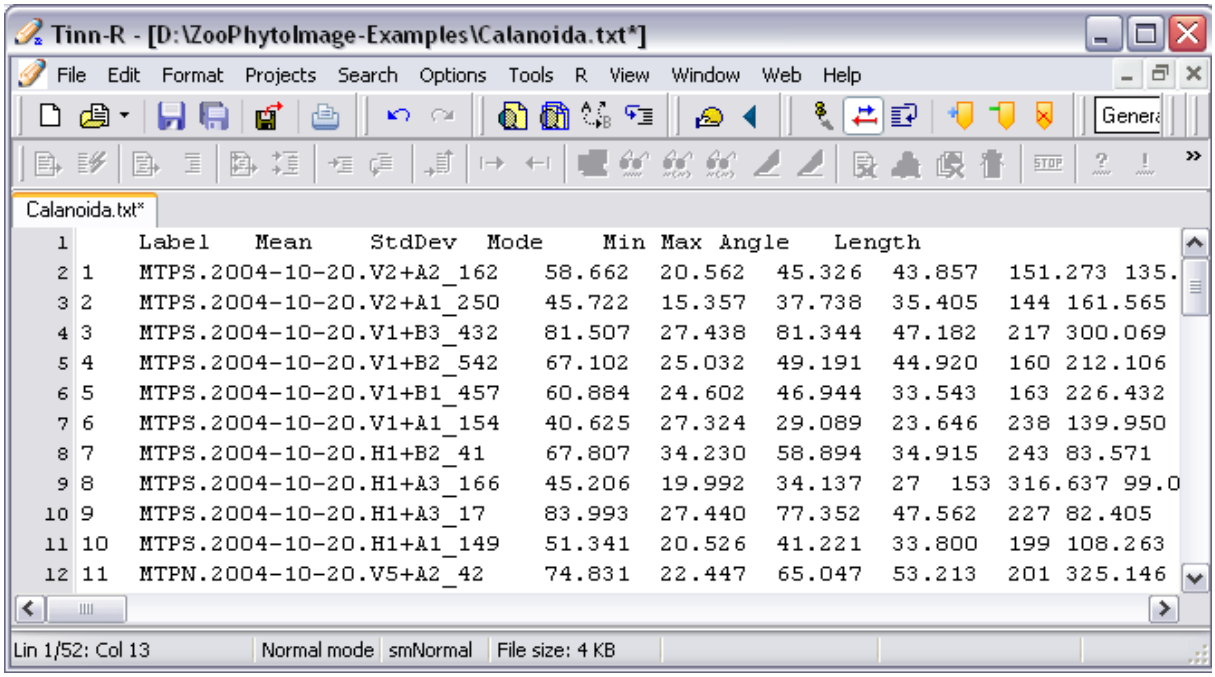


Manual measurement of "calanoida" prosome length in ImageJ, this parameter is usually measured under binocular.

After this step, you obtain a table of measurements made on all particles from the "calanoida" group of the training set.



The table of measurement can be saved as .xls file and then changed into txt file in order to obtain something like this:



The linear regression between “Length” manually measured and “ECD” automatically measured on particle during image analysis allows you to create a second relationship.

Length = 1.888 ECD ($r = 0.996$, $p < 0.001$).

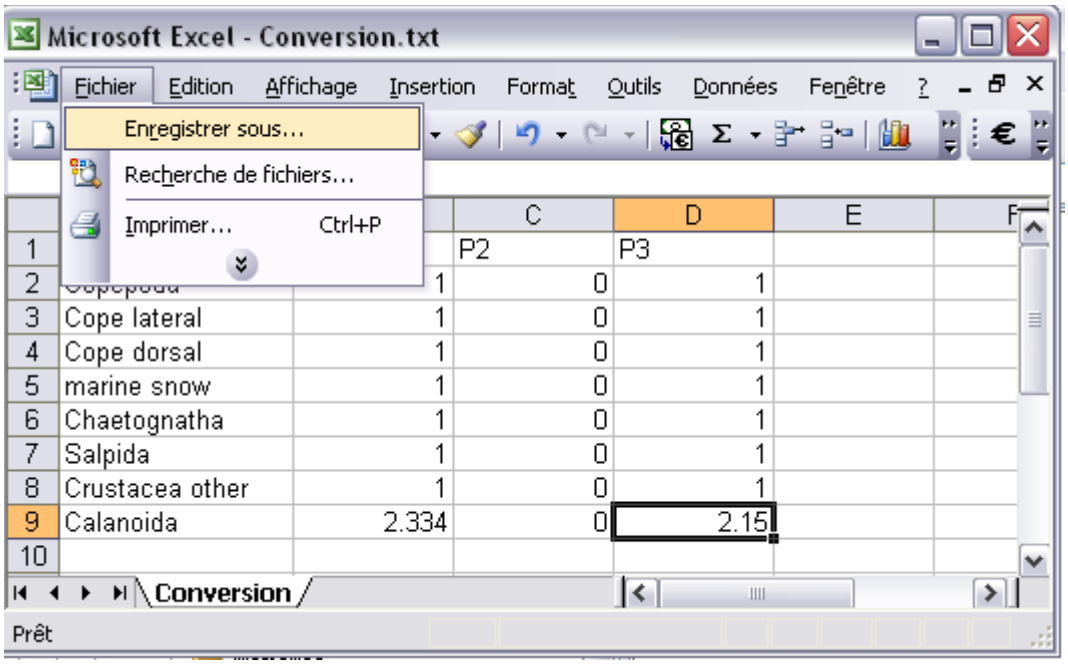
Do not forget to convert length into mm before calculating the linear regression.

Then, you have to replace the manual measurement of the literature by the automatic one, to obtain the final relationship used for biomass calculation.

$C = (2.334\ ECD)^{2.15}$

Pay attention, this indirect measurement of biomass introduces bias because it accumulates errors of calculation.

Finally, you have to fill in the conversion table with the different allometric parameters and then save it in the bin\R\R-2.4.1\library\zooimage\etc subdirectory of your installation directory. You can also create a table of conversion in a calculator file and save it as txt format in the adequate directory.



With this table, the software will calculate biomass for the "calanoida" group using a allometric relationship based on these parameters.

2. Calculating series

Having a ZIClass object in memory and conversion factors for biomass calculation, it is time to automatically recognize all particles of zid files and calculate biological and ecological parameters on them. First you need to created as many zis files as analysis of series you want to do. Zoo/PhytoImage uses your automatic recognition tool (ZIClass object) to automatically recognise all particles of zid files related in a given zis file. In the same time, it calculates abundances, biomasses and size spectra of each sample like explained before.

For size spectra calculation you have to specify the limits for the different size classes to consider. The default value creates a regular sequence from 0.25mm to 2mm with a class width of 0.1mm (**seq(0.25, 2, by = 0.1)**).

To process all samples in a given series, click on the tenth button on the Zoo/PhytoImage 1 assistant toolbar: 

… and select the corresponding .zis file. Zoo/PhytoImage calculates all parameters automatically using a ZIClass object.

Once the process is successful, yours results are saved in the ZIRes object. You can save this object on hard disk like ZItrain and ZIclass objects.

3. Visualizing results

Now you calculated your series, and generated a ZIRes object, you can graphically explore results like abundances, biomasses and size spectra (total or by taxa).

All graphs available in Zoo/PhytoImage are not publication-ready. They are exploratory graphs that allow you to have a general idea on phenomenon of interest. R provides much more analysis and graphic possibilities that you can see in this section, but to do so you have to practice R code.

Currently, the program proposes only a limited number of graphs and you cannot customize colors, titles, etc.). These graphs are sufficient for a rapid inspection of time series, but spatial components are not handled yet. Graphs in R are very flexible, and you can visualize your data in many other ways... It is just a limitation of the menu/dialog box interface of the current version of Zoo/PhytoImage that is so limiting!

To displays graphics of your results, click on the eleventh button on the Zoo/PhytoImage1 assistant toolbar: 

… and select the 'ZIRes' object you just created. The software allows you to select 1 to 12 graphs by R window for comparison.

For size spectra graphs, you have also to select a taxa. The graph shows the part of this taxa on total size spectra.

4. Exporting results

If, despite all the potentials of R to analyze your series in the current environment, you want to export data, you can do it easily. Click on the fore last button on the Zoo/PhytoImage1 assistant toolbar: 

The software saves results as tabulation-delimited ASCII files. They should be easy to read from any other software (Microsoft Excel, Matlab, etc). Zoo/PhytoImage creates one file for abundance-biomass results and one file by sample for size spectra results.

5. Analysing results in R

R provides a wide range of statistical methods to analyze your results. You can manage directly with R objects created during your experiment but also with table you export at the end of treatment. For example the package called PASTECS ("Package for Analysis of

Space-Time Ecological Series") is especially designed to analyse space-time series of ecological experiments like planktonic time series. With tools implemented in this package you can regularise tour series or separates seasonal pattern from the rest Take a look at PASTECS web site for more information (http://www.sciviews.org/pastecs).

Appendix

1. Data and Metadata in .zim files

Explanation of the different fields which have to be filled in zim files. This step can be done manually (see ScanCol24-example) or automatically by using a template .zie (see MicroCol24-example).

Entry	*Topic*	*Explanation*
ZI1	-	This is not an entry. It just tells it is a Zoo/PhytoImage1 file.
Author	Image	Who digitized the picture?
Hardware	Image	Device used to digitize the picture.
Software	Image	Acquisition software and version.
ImageType	Image	Type of image. For instance trans 16bits gray 2400dpi means image acquired in transparency of 16bit gray scales and a resolution of 2400dpi.
Code	Fraction	The same fraction identifier as in the file name A, B, etc.
Min	Fraction	Minimum mesh size used to retrieve this fraction in µm. Use -1 if none.
Max	Fraction	Maximum mesh size used to retrieve this fraction in µm. Use -1 for none.
SubPart	Subsample	Part of the sample that was digitized. If the picture contains only 10% of the organisms in your sample, SubPart = 0.1, for instance.
SubMethod	Subsample	Method used to get the part (volumetry, Motoda, Falsom, etc.)
CellPart	Subsample	Part of the cell containing the plankton that was actually digitized .
Replicates	Subsample	If you did replicated images with the same protocol for that fraction, how many replicates do you have? Note: Zoo/PhytoImage with average results among replicates instead of summing them.
VolIni	Subsample	The volume of seawater that was sampled in m3. This is required to calculate abundances and biomasses per m3.
VolPrec	Subsample	The precision on the sampled volume estimate in m3. This will be used for error evaluation (not implemented yet).

2. Data and Metadata in .zis files

Explanation of the different fields which have to be filled in zis files. Some fields have to completed and other not (see table). These fields are used to describe all your series.

Obligatory fields		
Key	***Section***	***Comment***
Id	Description	The short identify of the series.
Name	Description	A longer name for this series.
Description	Description	A short description of the series
Contact	Description	The name of a responsible person of this series.
Email	Description	The email address of the contact.

Label	Samples	The complete label of the sample, as in the file names.
Code	Samples	A code for this sample.
Date	Samples	The data of sampling (in yyyy-mm-dd format).
Latitude	Samples	The latitude of sampling (in +/-x.xx).
Longitude	Samples	The longitude of sampling (in +/-x.xx).

Optional fields		
Key	***Section***	***Comment***
Institution	Description	Name of the institution
Objective	Description	Summary of objectives
URL	Description	Link to a web site describing the series
Note	Description	Notes about series
Code	series	Code of series
Name	series	Complete name of series code
Project	series	Project of the series
Institution	series	Institution which commanded the series
Country	series	Country where the series is sampled
Location	series	Location of the sampling
Contact	series	Contact for information about the series
Email	series	Email address of the contact
URL	series	Link to a web site describing the series
Note	series	Note about the series
Code	Cruises	Code of the cruise
ShipName	Cruises	Name of the boat
ShipType	Cruises	Type of the boat
ShipCallSign	Cruises	Signalisation of the ship
PortDeparture	Cruises	Port of departure
PortReturn	Cruises	Port of arrival
Captain	Cruises	Name of the captain of the ship
Coordinator	Cruises	Name of the cruise's coordinator
Investigators	Cruises	Name of the investigator
Start	Cruises	Start point of the cruise
End	Cruises	End point of the cruise
SouthmostLat	Cruises	Southern latitude
WestmostLong	Cruises	Western longitude
NorthmostLat	Cruises	Northern latitude
EastmostLong	Cruises	Eastern longitude
Project	Cruises	Name of the project
URL	Cruises	Link to a web site describing the cruise
Note	Cruises	Note about cruise
Code	Stations	Code of the station
Location	Stations	Location of the sampled station
Latitude	Stations	Latitude of the sampled station
Longitude	Stations	Longitude of the sampled station
Start	Stations	Date of the starting of sampling
End	Stations	Date of the end of sampling
Frequency	Stations	Number of sampling by year
Depth	Stations	Depth of the station
Description	Stations	Description of the station
Note	Stations	Note about the station
SCS	Samples	SCS code of the sample
Series	Samples	Code of the series
Cruise	Samples	Name of the cruise
Station	Samples	Code of station

Time	Samples	Hour of sampling
TimeZone	Samples	Time zone relative to greenwich hour
CoordsPrec	Samples	Precision about location in km
Operator	Samples	Operator of the sampling
GearType	Samples	Type of gear
OpeningArea	Samples	Opening area of the net
MeshSize	Samples	Size of the net mesh
DepthMin	Samples	Minimal depth of sampling
DepthMax	Samples	Maximum depth of sampling
SampVol	Samples	Sampling volume
SampVolPrec	Samples	Precision about sampling volume
TowType	Samples	Type of tow
Speed	Samples	Speed of the boat in knot
Weather	Samples	Weather the day of sampling
Preservative	Samples	Type of preservation
Staining	Samples	Type of staining
Biovolume	Samples	Biovolume of the sample
Temperature	Samples	Temperature of sea water
Salinity	Samples	Salinity of sea water
Chla	Samples	Chlorophyll a concentration
Note	Samples	Note about the sample

Remember to send zis files, together with the other 'description' metadata to zooimage@sciviews.org, as it is stated in the Zoo/PhytoImage license! These information will be added to the Zoo/PhytoImage web site to track use of the software in the world.