



HAL
open science

Workflows and Components of Bioimage Analysis

Kota Miura, Perrine Paul-Gilloteaux, Sébastien Tosi, Julien Colombelli

► **To cite this version:**

Kota Miura, Perrine Paul-Gilloteaux, Sébastien Tosi, Julien Colombelli. Workflows and Components of Bioimage Analysis. Bioimage Data Analysis Workflows. Learning Materials in Biosciences. Springer, Cham, pp.1-7, 2020, 10.1007/978-3-030-22386-1_1 . hal-02735486

HAL Id: hal-02735486

<https://hal.science/hal-02735486v1>

Submitted on 21 May 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



Workflows and Components of Bioimage Analysis

*Kota Miura, Perrine Paul-Gilloteaux, Sébastien Tosi,
and Julien Colombelli*

- 1.1 Introduction – 2
- 1.2 Types of Bioimage Analysis Software – 2
- Bibliography – 6

What You Learn from This Chapter

Definitions of three types of bioimage analysis software—Component, Collection, and Workflow—are introduced in this chapter. The aim is to promote the structured designing of bioimage analysis methods, and to improve related learning and teaching.

1.1 Introduction

Software tools used for bioimage analysis tend to be seen as utilities that solve problems off-the-shelf. The extreme version of such is like: “If I know where to click, I can get good results!”. In case of gaming software, as the user gets more used to the software, the user can achieve the final stage faster. To some extent, this might be true also with bioimage analysis software, but there is a big difference. As bioimage analysis is a part of scientific research, the goal to achieve is not to clear the common final stage that everyone heads toward, but something original that others have not found out. The difficulty of the usage of bioimage analysis software does not only reside in the hidden commands, but also in the fact that the user needs to come up with more-or-less original analysis. Then, how can we do something original using tools that are provided in public?

In this short chapter, we define several terms describing the world of bioimage analysis software, which are “workflows”, “components”, and “collections”, and explain their relationships. We believe that clarifying the definition of these terms can contribute largely to those who want to learn bioimage analysis, as well as to those who need to design the teaching of bioimage analysis. The reason is that these terms link the generality of software packages provided in public, with the specificity and the originality of the analysis that one needs to achieve.

1.2 Types of Bioimage Analysis Software

Software packages such as ImageJ (Schneider et al. 2012),¹ MATLAB,² CellProfiler (Carpenter et al. 2006)³ or ICY (de Chaumont et al. 2012)⁴ are often used to analyze image data in life sciences. These software packages are “**collections**” of implementation of image processing and analysis algorithms. Libraries such as ImgLib2 (Pietzsch et al. 2012),⁵ OpenCV (Bradski 2000),⁶ ITK (Johnson et al. 2015a,b),⁷ VTK (Schroeder et al. 2006),⁸ and Scikit-Image (van der Walt et al. 2014)⁹ are also packages of image processing and analysis algorithms, although with a different type of user interface that is not graphical. We invariably refer to them as “**collections**”. To scientifically analyze and address an underlying biological problem, one needs to hand-pick some algorithms from these

-
- 1 ▶ <https://imagej.org>
 - 2 ▶ <https://nl.mathworks.com>
 - 3 ▶ <https://cellprofiler.org/>
 - 4 ▶ <http://icy.bioimageanalysis.org>
 - 5 ▶ <https://imagej.net/ImgLib2>
 - 6 ▶ <https://opencv.org>
 - 7 ▶ <https://itk.org>
 - 8 ▶ <https://vtk.org>
 - 9 ▶ <https://scikit-image.org>

collections, carefully adjust their functional parameters to the problem and assemble them in a meaningful order. Such a sequence of image processing algorithms with a specified parameter set is what we call a “**workflow**”. The implementations of the algorithms that are used in the **workflows** are the “**components**” constituting that workflow (or “**workflow components**”). From the point of view of the expert who needs to assemble a workflow, a collection is a package bundling many different **components**. As an example, many plugins offered for ImageJ are mostly also **collections** (e.g. Trackmate (Tinevez et al. 2016),¹⁰ 3D Suite (Ollion et al. 2013),¹¹ MosaicSuite¹²...), as they bundle multiple **components**. On the other hand, some plugins, such as Linear Kuwahara filter plugin,¹³ are a single **component** implemented as a single plugin.

Each **workflow** is uniquely associated with a specific biological research project because the question asked therein as well as the acquired image quality are often unique. This calls for a unique combination of **components** and parameter set. Some **collections**, especially those designed with GUI, offer **workflow templates**. These templates are pre-assembled sequences of image processing tasks to solve a typical bioimage analysis problem; all one needs to do is to adjust the parameters of each step. For example, in the case of Trackmate plugin for ImageJ (Tinevez et al. 2016), a GUI wizard guides the user to choose an algorithm for each step among several candidates and also to adjust their parameters to achieve a successful particle tracking **workflow** (see ► Chap. 4). When these algorithms and parameters are set, the **workflow** is built. CellProfiler also has a helpful GUI that assists the user in building a **workflow** based on **workflow templates** (Carpenter et al. 2006). It allows the user to easily swap the algorithms for each step and test various parameter combinations. ■ Figure 1.1 summarizes the above explanations.

Though such templates are available for some typical tasks, **collections** generally do not provide helpful clues to construct a **workflow**—choice of components to be used and approach taken to assemble those **components** depend on expert knowledge, empirical knowledge or testing. Since the biological questions are so diverse, the **workflow** often needs to be original and might not match any available **workflow templates**. Building a **workflow** from scratch needs some solid knowledge about the **components** and the ways to combine them. It also requires an understanding of the biological problem itself. Each **workflow** is in essence associated with a specific biological question, and this question together with the image acquisition setup affect the required precision of the analysis. For example, image data in general should not be analyzed at a precision higher than the physical resolution of the imaging system that captures those data.¹⁴ In some cases, a higher precision does not imply more meaningful results just because such precision can be irrelevant to the biological question. These aspects should be carefully considered during the planning of the analysis and the choice of the **components**, together with the choice of statistical treatment.

Many biologists feel difficulty in analyzing image data, because of the lack in skills and knowledge to close the gap between a **collection** of **components** and a practical

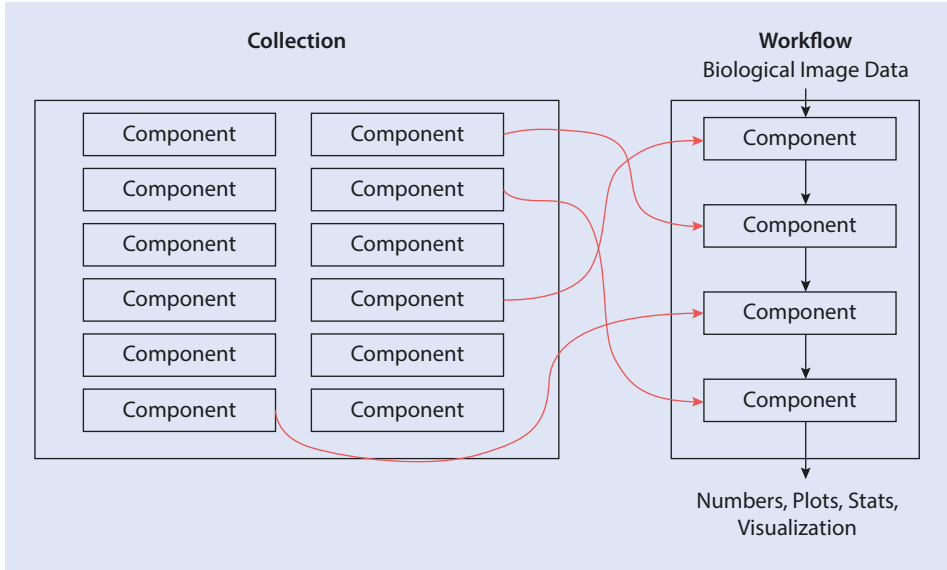
10 ► <https://imagej.net/TrackMate>

11 ► http://imagejdocu.tudor.lu/doku.php?id=plugin:stacks:3d_ij_suite:start

12 ► <http://mosaic.mpi-cbg.de/?q=downloads/imagej>

13 ► https://imagej.net/Linear_Kuwahara

14 If the model-based approach designed to compute sub-pixel resolution results is used e.g. single molecule localization microscopy, precision does go beyond the given optical resolution and the approach is thus validated.



■ Fig. 1.1 Relationship between components, collection and workflow. Components (e.g. Gaussian blurring filter) are selected from collection (e.g. ImageJ) and assembled into a specific workflow (red arrow) for analyzing image data in each research project (e.g. scripts associated with journal papers)

workflow. A **collection** bundles **components** without **workflows**, but it is often erroneously assumed that installing a **collection** is enough for solving bioimage analysis problem. The truth is that expert knowledge is required to choose **components**, adjust their parameters and build a **workflow** (■ Fig. 1.1 red arrows). The correct assembly of **components** as an executable script is in general even more difficult, as it requires some programming skills. The use of **components** directly from library-type of **collections**, which host many useful **components**, also requires programming skills to access their API. Bioimage analysts may fill this gap but even they, who professionally analyze image data, need to always search for the most suitable **components** to solve problems, reaching the required accuracy or coping with huge data in a practical time.

Another important aspect and difficulty is the reproducibility of **workflows**. We often want to know how other people have performed image analysis and to learn from others new bioimage analysis strategies. In such cases, we look for **workflows** addressing a similar biological problem. However, many articles do not document the **workflows** they used in sufficient details to enable the reproducibility of the results. As an extreme example, we found articles with their image analysis description in Materials and Methods merely documenting that ImageJ was used for the image analysis. Such a minimalism should be strictly avoided. On the other hand, some **workflows** are written as a detailed text description in Materials and Methods sections in the publications. We go even further and recommend to publish **workflows** as executable scripts, i.e. a computer program, with documented parameter sets for clarity and reproducibility of analysis and results. In our opinion, the best format is a version-tracked script because the version

used for the published results can be clearly stated and reused by others. A script embedded in a Docker image is even better for avoiding problems associated with a difference in execution environments.

Towards a more efficient designing of **workflows**, The Network of European Bioimage Analysts (NEUBIAS) has been developing a searchable index named Bioimage Informatics Search Engine (BISE). This service is accessible online at ► <https://biii.eu> and hosts the manually curated registry of **collections**, **workflows** and **components**.

Two ontologies are used for annotating resources registered to BISE: The BISE ontology for properties of resources e.g. programming language; and the EDAM Bioimaging Ontology (Kalaš et al. 2019)—an extension of the EDAM ontology (Ison et al. 2013) developed together with ELIXIR¹⁵—for applications of these resources, e.g. image processing step and imaging modality. “Component”, “Workflow” and “Collections” are implemented as part of the BISE ontology for classifying the type of software, for more distinctive filtering of search results.

While BISE allows researchers to search for bioimage analysis resources at all these levels, general web search engines, such as Google, typically return hits of **collections** but not to the details of their **components**. In addition, **workflows** are in many cases hidden in biological papers and difficult to be discovered. BISE is also designed to feature users impressions on the usability of **components** and **workflows** so that individual experiences can be swiftly shared within the community.

Take Home Message

Within the world of bioimage analysis software, various types of tools, which can be classified as “collections”, “components”, or “workflows”, coexist and are flatly provided to the public as “software tools”. Clear definition of these types and recognition of the role of each is a foundation for learning and teaching bioimage analysis.

■ ■ Further Readings

1. Miura and Tosi (2016) discusses the general challenges of bioimage analysis.
2. Miura and Tosi (2017) provides more details on the structure and designing of bioimage analysis workflows.
3. Details about NEUBIAS can be found at the following web pages:
 - ► <http://neubias.org>
 - ► <https://www.cost.eu/actions/CA15124>: The Memorandum of Understanding describes the objectives of the network, that includes the motivation to create the registry ► <http://biii.eu>.

Acknowledgements We are grateful to Nataša Sladoje for critically reading this text. We thank Matúš Kalaš for checking the text and correcting our mistakes.

15 ► <https://www.elixir-europe.org>

Bibliography

- Bradski G (2000) The OpenCV Library. Dr. Dobbs' Journal of Software Tools. <http://www.drdoobs.com/open-source/the-opencv-library/184404319>
- Carpenter AE, Jones TR, Lamprecht MR, Clarke C, Kang IH, Friman O, Guertin DA, Chang JH, Lindquist RA, Moffat J, Golland P, Sabatini DM (2006) CellProfiler: image analysis software for identifying and quantifying cell phenotypes. *Genome Biol* 7(10):R100. ISSN: 1465-6914. <https://doi.org/10.1186/gb-2006-7-10-r100>
- de Chaumont F, Dallongeville S, Chenouard N, Hervé N, Pop S, Provoost T, Meas-Yedid V, Pankajakshan P, Lecomte T, Le Montagner Y, Lagache T, Dufour A, Olivo-Marin J-C (2012) Icy: an open bioimage informatics platform for extended reproducible research. ISSN: 1548-7091
- Ison J, Kalaš M, Jonassen I, Bolser D, Uludag M, McWilliam H, Malone J, Lopez R, Pettifer S, Rice P (2013) EDAM: an ontology of bioinformatics operations, types of data and identifiers, topics, and formats. *Bioinformatics* 29(10):1325–1332. <https://doi.org/10.1093/bioinformatics/btt113>
- Johnson HJ, McCormick MM, Ibanez L (2015a) Template: the ITK software guide book 1: introduction and development guidelines, vol 1
- Johnson HJ, McCormick MM, Ibanez L (2015b) Template: the ITK software guide book 2: design and functionality, vol 2
- Kalaš M, Plantard L, Sladoje N, Lindblad J, Kirschmann MA, Jones M, Chessel A, Scholz LA, Rössler F, Dufour A, Bogovic J, Waithe CZD, Sampaio P, Paavolainen L, Hörl D, Munck S, Golani O, Moore J, Gaignardand A, Levet F, Participants in the NEUBIAS Taggathons, Paul-Gilloteaux P, Ison J, the EDAM Dev Team, Miura K, Colombelli J, Welcoming New Contributors (2019) EDAM-bioimaging: the ontology of bioimage informatics operations, topics, data, and formats (2019 update) [version 1; not peer reviewed]. *F1000Research* 8(ELIXIR):158. <https://doi.org/10.7490/f1000research.1116432.1>
- Miura K, Tosi S (2016) Introduction. In: Miura K (ed) *Bioimage data analysis*. Wiley-VCH, Weinheim, pp 1–3. ISBN: 978-3-527-34122-1
- Miura K, Tosi S (2017) Epilogue: a framework for bioimage analysis. In: *Standard and super-resolution bioimaging data analysis*. Wiley, Chichester, pp 269–284. <https://doi.org/10.1002/9781119096948.ch11>
- Ollion J, Cochennec J, Loll F, Escudé C, Boudier T (2013) TANGO: a generic tool for high-throughput 3D image analysis for studying nuclear organization. *Bioinformatics* 29(14):1840–1841. ISSN: 13674803. <https://doi.org/10.1093/bioinformatics/btt276>
- Pietzsch T, Preibisch S, Tomančák P, Saalfeld S (2012) ImgLib2—generic image processing in Java. *Bioinformatics* (Oxford, England) 28(22):3009–3011. ISSN: 1367-4811. <https://doi.org/10.1093/bioinformatics/bts543>
- Schneider CA, Rasband WS, Eliceiri KW (2002) NIH Image to ImageJ: 25 years of image analysis. *Nat Methods* 9(7):671–675 (2012). ISSN: 1548-7105. <http://www.ncbi.nlm.nih.gov/pubmed/22930834>
- Schroeder W, Martin K, Lorensen B (2006) *The visualization toolkit*, 4th edn. Kitware Inc. ISBN: 978-1930934191. <https://vtk.org/>
- Tinevez J-Y, Perry N, Schindelin J, Hoopes GM, Reynolds GD, Laplantine E, Bednarek SY, Shorte SL, Eliceiri KW (2016) TrackMate: an open and extensible platform for single-particle tracking. *Methods*. ISSN: 10462023. <https://doi.org/10.1016/j.ymeth.2016.09.016>. <http://linkinghub.elsevier.com/retrieve/pii/S1046202316303346>
- van der Walt S, Schönberger JL, Nunez-Iglesias J, Boulogne F, Warner JD, Yager N, Gouillart E, Yu T (2014) Scikit-image: image processing in Python. *PeerJ* 2:e453. ISSN: 2167-8359. <https://doi.org/10.7717/peerj.453>. <https://peerj.com/articles/453>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (► <http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

