QuPath: Open Software for Bioimage Analysis

Léo Leplat

22/05/2024

Peter Bankhead Research Group

The **Peter Bankhead Research Group** is a computational research group that develops methods and software to analyze **biomedical images**, led by **Peter Bankhead**.

Part of the group focus on **research**:

- Laura Nicolás-Sáenz (Wellcome Trust).
- Thibaut Goldsborough.
- Pau Carrillo Barberà.
- Kirti Prakash (Wellcome Trust).
- Hanane Issa.
- Oscar Forestier.

While the rest focus on **programming**:

- Fiona Inglis (Chan Zuckerberg Initiative).
- Alan O’Callaghan (Wellcome Trust).
- Léo Leplat (Chan Zuckerberg Initiative).

About me

- 2016-2018: Preparatory Classes (**Maths**, **Physics**).
- 2018-2020: Engineering School (various **engineering subjects**).
- 2020-2022: MSc in Simulation and Visualization (**Computer Science**).
- 2022-2023: Research Assistant (**Computer Graphics**).
- 2023- : Research Software Engineer (**Software development** applied to bioimage analysis).

QuPath: open-source software for bioimage analysis

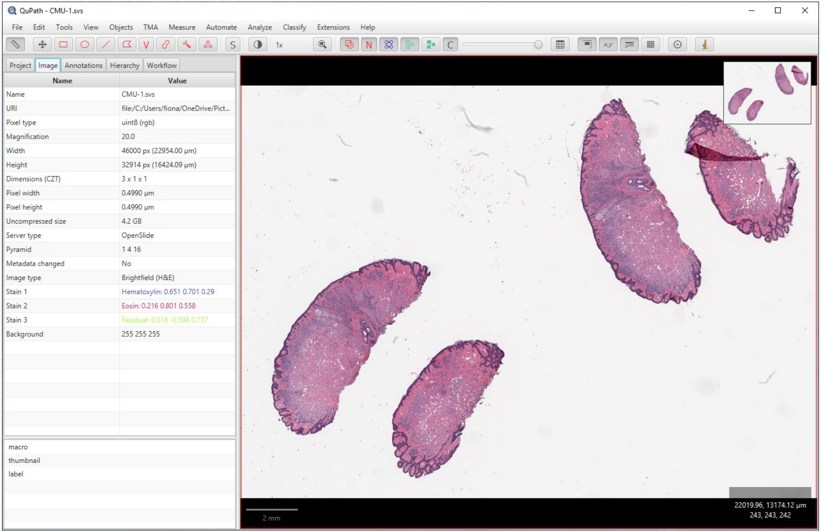
Our main focus is to develop **QuPath**.

QuPath is open-source software for **bioimage analysis**, often used for **digital pathology** applications because it offers a powerful set of tools for working with **whole slide images**.

It was created in 2017 and is used by a lot of people in the bioimage analysis world (around 40k downloads of the last version released two months ago).

Features include powerful annotation & visualization tools, built-in algorithms for common tasks such as cell and tissue detection, interactive machine learning...

QuPath: open-source software for bioimage analysis



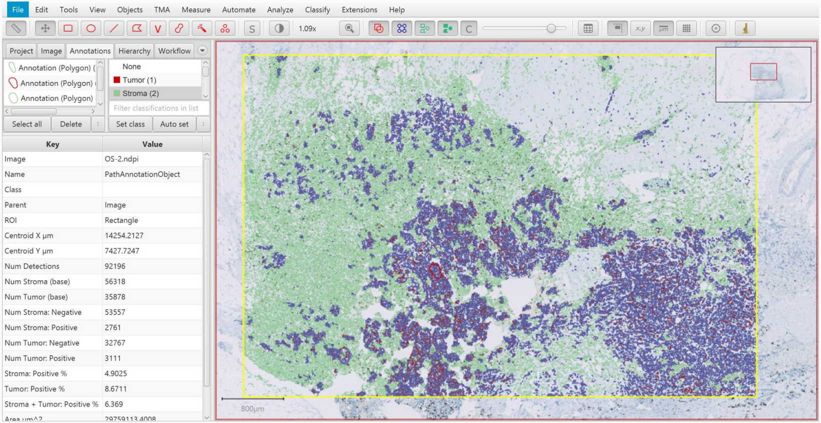
Name	Value
Name	CMU-1.svs
URI	file:/C:/Users/fiona/OneDrive/Pict...
Pixel type	uint8 (rgb)
Magnification	20.0
Width	46000 px (22954.00 µm)
Height	32914 px (16424.09 µm)
Dimensions (CZT)	3 x 1 x 1
Pixel width	0.4990 µm
Pixel height	0.4990 µm
Uncompressed size	4.2 GB
Server type	OpenSlide
Pyramid	1 4 16
Metadata changed	No
Image type	Brightfield (H&E)
Stain 1	Hematoxylin: 0.651 0.701 0.29
Stain 2	Eosin: 0.216 0.801 0.558
Stain 3	Residual: 0.316 -0.598 0.737
Background	255 255 255

macro
thumbnail
label

22019.96, 13174.12 µm
243, 243, 242

Figure 1: Overview of the software with an opened image

QuPath: open-source software for bioimage analysis



Key	Value
Image	OS-2.ndpi
Name	PathAnnotationObject
Class	
Parent	Image
ROI	Rectangle
Centroid X μm	14254.2127
Centroid Y μm	7427.7247
Num Detections	92196
Num Stroma (base)	56318
Num Tumor (base)	35878
Num Stroma: Negative	53557
Num Stroma: Positive	2761
Num Tumor: Negative	32767
Num Tumor: Positive	3111
Stroma: Positive %	4.9025
Tumor: Positive %	8.6711
Stroma + Tumor: Positive %	6.369
Area μm^2	29759113.4008

Figure 2: Cell classification results of an image

QuPath software stack

QuPath is entirely written in **Java**, because:

- Other open-source digital pathology tools (*ImageJ*, *Fiji*) are written in Java, so there is a high **interoperability** between them.
- Java is **easy to use** and provides a **lot of APIs**, and it is **cross-platform**.
- Java makes it easy to build **robust software** (OOP, Javadoc).
- The **Gradle** build tool makes it very easy to set up a developing environment.

However, Java is quite verbose, and it's difficulty to communicate with Python.

QuPath development

Part of the team focus on **research** (e.g. new cell segmentation algorithms using Machine Learning), while the rest focus on **software development** (fixing bugs, improving UI, adding new features).

Most of the time, each developer works on a **different task**. When a task is done, a GitHub **pull request** is made and others **check** it.

QuPath development

We follow a **style guide** to have a consistent syntax across the entire codebase. This guide simply consists of a GitHub project with a *README* file and includes:

- **Coding standards**, e.g. naming conventions.
- **Best practices**, e.g. function lengths.

We read **books** and then make **presentations** to the rest of the team. For example, I recently made a presentation on concurrency after reading the *Java Concurrency in practice* book.

We strive to make the code as **clean** as possible, because it reduces the complexity of the software and thus make it easier to **debug** and add **new features**. This can be done with:

- Following **best practices**.
- Having a **consistent code style**.
- Implementing **unit tests**.

QuPath user experience

We have a **ReadTheDocs** website that provides **documentation** for **using QuPath**. There is usually a **text** explanation and a **video** on each page. Topics covered include:

- Installing the software.
- How to use QuPath.
- Some step by step tutorials.
- Some advanced topics...

The **image.sc forum** (forum about scientific imaging) is the place where users can **ask questions** about the software, for example:

- How to use a feature.
- Ask for a new feature.
- Report a bug.

Recently, one member of our team (Laura) started a series of **video tutorials** on QuPath.

QuPath user experience

We have a **GitHub page** that provides useful information:

- QuPath **download** links and links to the **documentation** (for new users).
- Link to the **forum** (for users having a question).
- Link to the **GitHub project** (for developers).

One person of the group (Fiona) is mainly focusing on the **user interface** so that the software is easy to use for non-programmers.

QuPath user implication: scripting

Sometimes, the user interface of QuPath cannot easily provide a feature (e.g. batch processing). For these cases, an integrated **script editor** allows the user to create **scripts** to automatize processes.

Some users **share** their custom **script** with the community on the forum, which is useful to other users. This also helps seeing the direction the user interface development should take (ideally, the UI is sufficient for every cases).

QuPath user implication: extension

For bigger features, QuPath also allows users to develop **extensions**. A template provides an easy way to start the development of an extension.

Once an extension is developed, other users can easily **drag and drop** the built file to QuPath to **install** it.

This resulted in a quite a few extensions developed outside of the QuPath team and used by a lot of people.

QuPath recognition

We ask people to cite the QuPath paper if they use the software in their work.

Sometimes, we help some research groups to develop algorithms or to integrate them to QuPath. In these cases, members of our team are included in the corresponding papers.