https://doi.org/10.1038/s41592-023-01811-4

PoCA: a software platform for point cloud data visualization and quantification

Single-molecule localization microscopy (SMLM) is a mature but growing field that still lacks an efficient and user-friendly analysis and visualization software platform adapted for both users and developers. This is especially true when dealing with 3D quantification and large datasets, with only a couple of recent software platforms capable of properly handling 3D SMLM data[1,2]. We here present Point Cloud Analyst (PoCA), open-source software designed to ease the manipulation and quantification of multidimensional SMLM point cloud data (Fig. 1a). PoCA is the legacy of SR-Tesseler and Coloc-Tesseler[3,4]. Driven by community needs as well as the evolution of SMLM acquisition and analysis techniques, PoCA integrates the following key features:

Usability

The PoCA graphical user interface (GUI) is simple, intuitive and user friendly (Fig. 1b). It is designed to allow users as well as developers to perform efficient investigation and quantitative analysis of SMLM data on a standard desktop computer equipped with a graphics processing unit (GPU). Every action or plot is available in a few mouse clicks, improving user efficiency and visual feedback. Filtering, object creation, data interaction and quantifications are examples of available native functionalities. Each computed quantitative feature (for example, localization density or object characteristics) can be displayed to improve visual feedback. PoCA also integrates capabilities such as file drag and drop, camera saving or look-up-tables.

Visualization

PoCA is built around a custom Open-GL-based rendering engine that provides full user interactive control of SMLM point cloud data. It integrates several advanced visualization modes made to improve data comprehension. This includes uncertainty-based Gaussian intensity rendering (Fig. 1c), screen space ambient occlusion (Fig. 1d) and heat maps. 3D raw and segmented data can be manipulated for deeper investigations, thanks to annotated object bounding boxes and data cropping (Fig. 1c).

Efficiency and interactivity

PoCA exploits the GPUs for fluid interactive visualization and manipulation of millions of localizations, in two and three dimensions. It integrates optimized hybrid CPU/GPU algorithms for Voronoi diagram generation, allowing the reconstruction of 4,000,000 localizations in less than 35 s in 2D and 80 s in 3D on a regular workstation equipped with a NVIDIA graphic card. We also integrated an optimized version of the computationally demanding method DBSCAN[5], allowing the segmentation of 1,600,000 3D localizations in 80 s.

Modularity

PoCA relies on a modular design composed of a backbone and a plug-in architecture facilitating the addition of functionalities (Fig. 1e and Supplementary Note). This plug-in architecture provides the developers with the capacity to develop analysis methods in C++, as well as adding widgets to the GUI.

Automation and batch processing

Every action that can be performed in PoCA is internally described by a JavaScript Object Notation (JSON) file, a concise ASCII file format. It allows recording and displaying each command run by PoCA in a user-friendly macro recorder interface for creating and executing macros, an important feature for automation, batch analysis and reproducibility (Fig. 1e).

Advanced point cloud data visualization and manipulation

The CGAL library (https://doc.cgal.org/5.5.1/Manual/packages.html) integrated into PoCA allows advanced geometric computations, such as the kd-tree, octree or Voronoi diagram, to be performed efficiently. It also provides several data structures, such as meshes, polyhedrons or tetrahedrons, allowing segmented objects to be rendered as surface (mesh) or volume (tetrahedral) models and quantified (Fig. 1d). CGAL also provides advanced functionalities, such as reconstructing Poisson surface from point clouds or checking the belonging of a point inside an object, as required, for example, to compute object colocalization.

Python code integration

While developed in C++, PoCA integrates a native Python code interpreter through the NumPy library[6], allowing bidirectional transfers of arrays from C++ to the Python interpreter (Fig. 1e and Supplementary Note). Python code can then be executed on those arrays and the results carried to PoCA for visualization or further analysis. This allows rapid code prototyping, as well as allowing advanced computational methods, such as machine learning algorithms[7], to be executed in the PoCA environment (Supplementary Note and Supplementary Fig. 1).

To conclude, PoCA is a flexible software platform dedicated to the analysis and visualization of 2D and 3D localization data. It combines the strengths of both the C++ and Python programming languages, providing access to efficient and optimized C++ computer graphics algorithms and the Python ecosystem. It is designed to improve users' and developers' experience by integrating a user-friendly GUI with an efficient 3D rendering engine, a macro recorder and the capability to execute Python code. PoCA can also be used with any kind of point cloud, such as Protein Data Bank files[8]. Future directions include the integration of analysis methods for qPAINT[9], cluster distribution analysis on 3D surfaces and single-molecule tracking.

Code availability

PoCA is packaged as a one-click installer for the Windows operating system. The source code is available on GitHub (https://github.com/flevet/PoCA) under a LGPL v3 license. We provide a cmake script to facilitate its building inside other operating systems. The documentation is available at https://poca-smlm.github.io/.

Florian Levet [1,2] &
Jean-Baptiste Sibarita [1]
[1]Univ. Bordeaux, CNRS, Interdisciplinary Institute for Neuroscience, Bordeaux, France.
[2]Univ. Bordeaux, CNRS, INSERM, Bordeaux Imaging Center, UAR3420, US 4, Bordeaux, France.
e-mail: florian.levet@inserm.fr; jean-baptiste.sibarita@u-bordeaux.fr

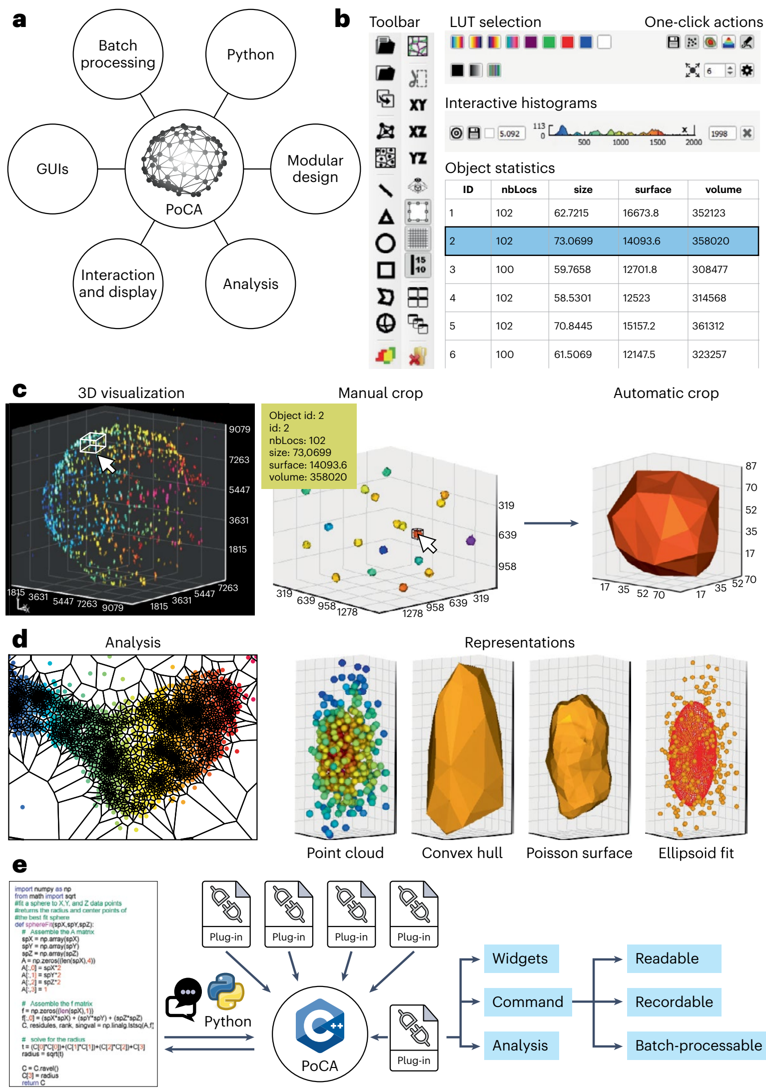

Fig. 1 | Overview of PoCA. **a**, The main features of PoCA. **b**, Snapshots of some GUIs allowing rapid visualization of quantitative features. LUT, look-up table. **c**, Examples of visualization and manipulation of 3D SMLM data. Left, uncertainty-based Gaussian intensity rendering. Center, manual cropping of white box at left. Quantitative features of the selected object are displayed in the yellow box. Right, automatic cropping of segmented data obtained by double-clicking an object of interest. **d**, Different object representations from which specific quantitative features are computed, including, from left to right, 2D Voronoi diagram, 3D point cloud with density coding, 3D convex hull, 3D Poisson surface and 3D ellipsoid fitting. **e**, Global architecture of PoCA, allowing plug-in development, batch processing and communication with Python.

Published online: 3 March 2023

References

1. Ries, J. *Nat. Methods* **17**, 870–872 (2020).
2. Marin, Z. et al. *Nat. Methods* **18**, 582–584 (2021).
3. Levet, F. et al. *Nat. Methods* 12, 1065–1071 (2015).
4. Levet, F. et al. *Nat. Commun.* **10**, 2379 (2019).
5. Ester, M., Kriegel, H.-P., Sander, J. & Xu, X. In *Proc. Second Int. Conf. Knowledge Discovery and Data Mining (KDD'96)* 226–231 (AAAI Press, 1996).
6. Harris, C. R. et al. *Nature* **585**, 357–362 (2020).
7. Williamson, D. J. et al. *Nat. Commun.* **11**, 1493 (2020).
8. Berman, H. M. et al. *Nucleic Acids Res.* **28**, 235–242 (2000).
9. Jungmann, R. et al. *Nat. Methods* **13**, 439–442 (2016).

Acknowledgements

This work was supported by the Ministère de l'Enseignement Supérieur et de la Recherche (ANR-20-CE11-0006 NANO-SYNATLAS to F.L.), the Centre National de la Recherche Scientifique (CNRS), the Conseil Régional d'Aquitaine and the Institut National de la Santé et de la Recherche Médicale (Inserm). We also acknowledge France-BioImaging infrastructure supported by the French National Research Agency (ANR-10-INBS-04).

Author contributions

F.L. originated the idea and developed the software. F.L. and J.-B.S. designed the software requirements and contributed to the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41592-023-01811-4.

Peer review information *Nature Methods* thanks Ann Wheeler and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.