Tomography analysis tool: an application for image analysis based on unsupervised machine learning

T Bagni, H Haldi, D Mauro and C Senatore

1 Department of Quantum Matter Physics, University of Geneva, Geneva, Switzerland
2 Department of Computer Science, University of Geneva, Geneva, Switzerland
E-mail: tommaso.bagni@unige.ch

Abstract
We developed a graphical user interface (GUI) to analyse tomographic images of superconducting Nb₃Sn wires designed for the next generation accelerator magnets. The Tomography Analysis Tool (TAT) relies on the k-means algorithm, an unsupervised machine learning technique which is widely used to partition images into separated clusters. The GUI is compatible with both Linux and Windows operating systems. The software reliability was tested by optical inspecting the tomographic images superimposed on the clustered image obtained by the k-means algorithm. TAT was proven to correctly segment the various components of the Nb₃Sn superconducting wires with single pixel precision. Finally, this software can be a useful tool for the scientific community to segment and analyse quickly and reproducibly tomographic images.

Introduction

Nb₃Sn is an isotropic intermetallic compound with an A15 crystal structure [1]. Its critical temperature (Tc) is \( \sim 18 \) K, and its upper critical field (Bc2) can reach \( \sim 30 \) T at 0 K [2]. These characteristics combined with a high critical current density \( (J_c) \) make Nb₃Sn the ideal material for superconducting magnets operating between 10 T and 20 T. Therefore, it is the ideal candidate for the 16 T dipole magnets that CERN is designing for its new proton-proton collider, the so-called Future Circular Collider (FCC) [3].

Nb₃Sn-based magnets are prepared following the wind & react approach [4]. Unlike ductile Nb-alloys, Nb₃Sn is brittle and strain sensitive intermetallic compound, which cannot be directly drawn in the form of a wire. Instead, ductile precursor components are combined, drawn to form a wire, brought to final shape (e.g. a coil) and then heat treated so that Nb₃Sn forms from the precursors in a reactive diffusion process. One of the most promising Nb₃Sn technology is the Internal-Tin [5], which consists of ring-like arrays of Nb-alloy filaments inside a Cu matrix with a Sn core inside each of these rings, often called sub-elements. The Cu surrounding the Nb-alloy filaments provides a pathway for a rapid diffusion of the Sn to form the Nb₃Sn phase during the reaction heat treatment. Sub-elements are enclosed by a Nb or Ta diffusion barrier to prevent Sn poisoning during the reaction of the Cu matrix embedding the sub-elements, which acts as thermal stabilizer [6, 7].

In addition to the Nb₃Sn phase, the reaction heat treatment generates Kirkendall voids [8] in the wire sub-elements that can have a detrimental impact on mechanical performances of the wires [9], see figure 1 for an example of Nb₃Sn wire cross section. As said, Nb₃Sn is a brittle and strain sensitive compound, this imply that it is prone to fracture and its critical current density strongly depends on the mechanical strain applied to the superconducting wire. In magnet design, the action of Lorentz forces generated by high current and high magnetic fields results in mechanical loads which translate into a reduction of the conductor performance [10]. In this context, to further improve Nb₃Sn technology and achieve the performance targets in view of the FCC [11], it is necessary to fully understand the impact such voids have on the strain sensitivity of different wire cross sections.

3 The wire cross section was collected at the University of Geneva by low angle backscatter scanning electron microscopy (LABE-SEM) with beam energy at 20 kV.
designs. A fundamental step on this path is the 3D reconstruction of the wire geometry, which is necessary to create a Finite Element Models (FEM) at sub-elements level, that would allow further optimization of electromagnetic properties of Nb$_3$Sn wires. To reach this goal, we developed a tool to characterize completely the internal structure of Nb$_3$Sn wires applying unsupervised machine learning to x-ray micro-tomography [12].

Synchrotron tomography is the perfect technology to study in a non-destructive way the internal features of a dense object, and it was used successfully in the past to study and characterize the voids formation in Internal-Tin Nb$_3$Sn wires [13, 14].

The Tomography Analysis Tool (TAT) is a Python application that enables a user-friendly post-process of tomographic images based on $k$-means, i.e. the same unsupervised machine learning used in [12]. TAT exploited $k$-means to divide the different components of a wire in separate clusters, based on the brightness and colour of the corresponding pixels, and save them as separate images and matrices compatible with other softwares, e.g., MATLAB, Python or ImageJ. $K$-means is particularly effective when the components are characterized by different pixel colour or brightness. Furthermore, TAT is made to study single images as well as large dataset ($\sim$1000 images) and is compatible with the most common image formats as PNG, JPG and TIFF. Even if the application was developed for analysing Nb$_3$Sn wires, the flexibility of $k$-means allows to use TAT for every application which requires image clustering.

The paper first discusses the $k$-means algorithm and TAT workflow. The second part is dedicated to TAT interface, functionality and possible applications.

The tomographic images used as case study for TAT features were acquired at the micro-tomography beamline ID19 of the European Synchrotron Radiation Facility (ESRF), France, with an average beam energy of 89 keV and a resolution 0.7 $\mu$m pixel$^{-1}$.

**Methods**

**k-means**

The separation of the various components in an image can be done using the $k$-means algorithm, an unsupervised classification algorithm developed for clustering [15]. The advantage of unsupervised learning is that it exhibits self-organization and, in the case of $k$-means, captures patterns as probability densities [16], thus having the capability to operate without training. The intuition behind clustering is to divide a given set of data into a specific number of groups based on patterns or similarities present in the data; in our case the groups (e.g., sub-elements, Cu matrix and voids, see figure 1) are defined based on the pixel brightness in the tomographic image.
The algorithm divides the given set of data into \( k \) disjoint clusters matching the number of the chosen groups. The analysis of an image, e.g. one of the slices from a tomography or a SEM picture, can be summarized as follows: in the initial step \( k \)-means generates a \( k \) number of centres, \( c_k \), based on the pixel brightness scale of the picture. The initial centres are arbitrarily generated by the algorithm, while the operator can control the number \( k \) of clusters, the number of times the algorithm will run with different initial \( c_k \), and the maximum number of iteration per run \([17]\). For each pixel of the image, the Euclidean distance \( d \), between the centres and the pixel in the brightness scale, \( p(x, y) \), is calculated as:

\[
d = |p(x, y) - c_k|
\]  

(1)

then, each pixel is assigned to the nearest center based on the calculated distance \( d \). When all pixels have been assigned, new centers are generated using equation (2):

\[
c_k = \frac{1}{n_k} \sum_{y \in c_k} \sum_{x \in c_k} p(x, y)
\]  

(2)

where \( n_k \) is the number of observations in the \( k \)th cluster \([18]\). Then, the process is repeated until there are no significant changes in the centers positions.

As a last step the clusters are saved, and the clustered image is generated. More on \( k \)-means algorithm can be found in \([19]\). \( k \)-means algorithm was implemented in TAT using the python scikit-learn library \([20]\).

**App workflow and verification**

The TAT process can be divided into two main stages as shown in the workflow depicted in figure 2. The first part (left side of figure 2) is focused on image clustering, i.e., finding a proper number of clusters by applying the \( k \)-means algorithm using equations (1) and (2). The program allows the user to select the input images directory and the output destination directory. The user can select a specific set of images among the uploaded ones and define the \( k \)-means parameters. The output images can be readily controlled and evaluated inside the application once the clustering of all the selected images is over (we report an average processing time per image between \( \sim 2\text{–}8 \text{s} \)) and, if necessary, the procedure can be rapidly repeated changing the parameters. All the outputs are automatically saved in the dedicated folder at every iteration.

In some cases, and especially in the data exploration phase, this part of the program is sufficient already. On the other hand, in case of poor image quality or if some features are not correctly clustered, it can be beneficial to use a number of clusters exceeding the number of the researched features. These clusters can be then combined

---

3 Time obtained running several times TAT on a batch of 20 images 512 × 512 pixels on an intel i7 personal laptop. The range in time is due to the change of the number of clusters ranging from 4 until 10.
to increase the final cluster precision. For this purpose, the second functionality of TAT (right side of figure 2), referred as 'merging tool', is designed to merge different clusters together. This option allows to combine different components into a larger cluster whenever is necessary to the image analysis, e.g. to compensate for image aberrations. Completed the initial clustering (part one), the user can select which clusters to merge. After evaluating the result, the user can directly apply the merging to all the processed images, or if not satisfied it is possible to 'undo' the merge and modify the cluster aggregation.

**TAT interface**

TAT is designed to run both on Windows and on Linux operating systems with an identical GUI. Figure 3 shows the main window of TAT where the user can start the analysis of the tomographic images. Two buttons are dedicated to select the input directory for uploading the images, and the output directory for saving. Once the images are loaded, they can be seen as a small preview in the bottom panel of the interface (see panel 1 of figure 3). The user can choose which images are going to be processed selecting them by using the ticking box. There is a dedicated 'select/deselect all' button to simplify and speed up the image selection process. The user can input the desired number of clusters in which the images will be partitioned using the 'Cluster count' tab (green box). Additionally, we added a 'Run count' tab (red box) in order to define how many clustering attempt the software will run, and a 'Max iterations' tab (blue box) used in case the algorithm fails to converge or if the user wants to extend the number of iterations. Once these values are set the process of generating the clusters can be launched with the 'Generate' button.

At the end of the clusters generation, the clustered images appear as a preview on the top right column of the main window, see panel 2 of figure 3. Each output can be examined by clicking on the respective preview, which will be shown in the main panel (panel 3 of figure 3). In case the result is not satisfying the user can reset TAT by clicking on the 'Clear' button, changing the parameters and repeating the process. Once the ideal conditions are found, the user can either stop the analysis, since at every run the images and all the clusters matrices are saved in the output folder, both as PNG images and NumPy arrays (*.npy file), or proceed to the merging phase in case further processing is needed.

As it is often the case, in order to have higher precision and gather more information from the various components, the capability of using more clusters turns out to be extremely useful. In fact, more clusters allow a more detailed pixel-wise partition of the components of an image making appear features that otherwise would not be partitioned correctly. For this reason, we created a separated window for the second stage of the analysis called 'Cluster editor' that can be accessed by double clicking on any of the clustered miniatures on the right column of the interface.
Figure 4 shows the interface of the Cluster Editor window, in this example, the same image showed in figure 3, i.e. ‘image_slice_0504’, is analysed using 15 clusters, instead of 4, to better define sub-element edges. The central panel shows a single cluster (panel 2 of figure 5), which can be selected by clicking on the clusters miniatures in the right column. The merging can be done by selecting multiple layers followed by pressing the ‘Merge’ button. In case the resulting combination is not ideal, it is possible to use the ‘Reset’ button to start the procedure again.

The final merged image appears in figure 5, where the 15 clusters are merged into 3 (the image colour code changes as function of the number of clusters). Once the layers are successfully merged, it is possible to apply the same changes to all the other images that were analysed using the ‘Apply to all’ button. This will save all the new merged images inside the ’merged’ sub-folder that TAT previously created in the output directory. Applying the merging to all the images, the software imposes the same colour code through all the image clusters, this is extremely useful in further post-processing steps where the single components can be retrieved by simply selecting the corresponding pixel value which is now the same along all the frames. To be noted, in case of bad

Figure 4. TAT merging interface. Panel 1 shows clustered image with each cluster defined by a different colour. Panel 2 shows the cluster 14 as binary image. In the right column the control buttons and the clusters miniatures are listed. In this specific case, the input image ‘image_slice_0504’ was segmented in 15 clusters (from 0 to 14).

Figure 5. TAT merging interface after merging the clusters. The 15 clusters (from 0 to 14) of figure 4 were merged in three new clusters, as indicated in the miniature names in the right column. Panel 1 shows the clustered image with a different colour for each cluster, and panel 2 shows the selected cluster as binary image.

Note that during the cluster calculation, the color of the clusters can slightly vary between one image and the other. This ends up having different color-coded images for same components and might complicate the analysis. The ‘merging’ procedure takes care of this detail.
merging instead of resetting the entire picture it is also possible to ‘undo’ the last merge by clicking the dedicated ‘Undo’ button.

The verification of TAT outputs was done upon visual inspection of the output images. Currently there are not known applications readily available to separate the components in the images of a superconducting wire exploiting the k-means algorithm. Therefore, the goodness of the result was evaluated observing the 3D reconstruction of several Nb₃Sn superconducting wires and analysing their voids distribution [12]. Following the 3D reconstruction, it was possible to evaluate if any component was missing in the analysed images (approximately 500 images for every batch). An example of 3D reconstruction is shown in figure 6.

Furthermore, even if the TAT was developed to analyse superconducting wires, it can have significant use also in other application which requires image clustering. The robustness of the k-means algorithm is well known to the research community [21, 22], in particular in segmentation problems for biology [23] and medical applications [24], making TAT an efficient and ideal tool to analyse images for a broad spectrum of researchers. Finally, TAT can be exploited to generate the first training datasets for more sophisticated supervised artificial intelligences, like Convolutional Neural Networks (CNNs), speeding up the training process of learning networks.

More examples of analyses performed using unsupervised machine learning applied to superconducting wires can be found in [12] [25].

Conclusions

We have developed the Tomography Analysis Tool, an open-source user-friendly application to post-process efficiently tomographic images of superconducting wires exploiting the k-means algorithm. TAT allows a very quick and precise identification of all the components of a wire. This tool was specifically developed to help analyze superconducting wires through image analysis but could also be applied to different complex systems as biological or medical images. TAT can be quickly deployed for other specific user-cases wherever it is necessary to separate an image into clusters.

Tomography Analysis Tool can be downloaded at: https://tat.readthedocs.io.

Acknowledgments

This work was done under the auspices of CHART (Swiss Accelerator Research and Technology Collaboration, https://chart.ch).

Financial support was provided by the European Organization for Nuclear Research (CERN), Memorandum of Understanding for the FCC Study, Addendum FCC-GOV-CC-0176 (KE4612/ATS).
Research also supported by the European Synchrotron Radiation Facility (Grant No. MA-2767).

**Data availability statement**

The data that support the findings of this study are openly available at the following URL/DOI: [https://doi.org/10.26037/yareta:noxmqspsnznxckm5f2kgajo3vy](https://doi.org/10.26037/yareta:noxmqspsnznxckm5f2kgajo3vy).

**ORCID iDs**

T Bagni [https://orcid.org/0000-0001-8654-783X](https://orcid.org/0000-0001-8654-783X)
D Mauro [https://orcid.org/0000-0002-8290-1165](https://orcid.org/0000-0002-8290-1165)
C Senatore [https://orcid.org/0000-0002-9191-5016](https://orcid.org/0000-0002-9191-5016)

**References**

[1] Matthias B T, Geballe T H, Geller S and Corenzwit E 1954 Superconductivity of Nb3Sn *Phys. Rev.* 95 1435–1435
[2] Foner S and McNeill E J 1981 Upper critical fields of cubic and tetragonal single crystal and polycrystalline Nb3Sn in DC fields to 30 tesla *Solid State Commun.* 39 959–64
[3] Abada A et al 2019 FCC-hh: the Hadron Collider *Eur. Phys. J. Spec. Top.* 228 755–1107
[4] Ishibashi K, Koizumi M, Hosoyama K, Kobayashi M and Horigami T 1981 Nb3Sn dipole magnet by wind and react process *IEEE Trans. Magn.* 17 428–31
[5] Hashimoto Y, Yoshizaki K and Tanaka M 1974 *Processing and Properties of Superconducting Nb3Sn Filamentary Wires* (United Kingdom: IPC Business Press)
[6] Suenaga M, Tsuchiya K and Higuchi N 1984 Superconducting critical current density of bronze processed pure and alloyed Nb3Sn at very high magnetic fields (up to 24 T) *Appl. Phys. Lett.* 44 919–21
[7] Tarantini C, Lee P J, Craig N, Ghosh A and Larbalestier D C 2014 Examination of the trade-off between intrinsic and extrinsic properties in the optimization of a modern internal tin Nb3Sn conductor *Supercond. Sci. Technol.* 27 065013
[8] Easton D and Kroeger D 1979 Kirkendall voids—a detriment to Nb3Sn superconductors *IEEE Trans. Magn.* 15 178–81
[9] Barth C et al 2018 Quantitative correlation between the void morphology of niobium-tin wires and their irreversible critical current degradation upon mechanical loading *Sci. Rep.* 8 6589
[10] Bordini B, Alknes P, Bottura L, Rossi L and Valentinis D 2016 An exponential scaling law for the strain dependence of the Nb3Sn critical current density *Supercond. Sci. Technol.* 28 075014
[11] Tommasini D et al 2017 The 16 T dipole development program for FCC *IEEE Trans. Appl. Supercond.* 27 1–5
[12] Bagni T et al 2021 Machine learning applied to x-ray tomography as a new tool to analyze the voids in RRP Nb3Sn wires *Sci Rep.* 11 7767
[13] Scheuerlein C, Di Michiel M and Haibel A 2007 On the formation of voids in internal tin Nb3Sn superconductors *Appl. Phys. Lett.* 90 132510
[14] Scheuerlein C, Di Michiel M and Buta F 2009 Synchrotron radiation techniques for the characterization of Nb3Sn superconductors *IEEE Trans. Appl. Supercond.* 19 2653–6
[15] MacQueen J 1967 Some methods for classification and analysis of multivariate observations Proc. of the Fifth Berkeley Symp. on Mathematical Statistics and Probability vol 1 (Berkeley, Calif.: University of California Press) 281–97
[16] Hinton G E and Sejnowski T J 1999 *Unsupervised Learning: Foundations of Neural Computation* (Cambridge, Massachusetts: MIT press) (https://doi.org/10.7551/mitpress/7011.0001.0001)
[17] Arthur D and Vassilvitskii S 2006 *Proceedings of the Twenty-Second Annual Symposium on Computational Geometry* (Sedona, Arizona, USA: Association for Computing Machinery) 144–53
[18] James G, Witten D, Hastie T and Tibshirani R 2013 *An Introduction to Statistical Learning* vol 112 (New York: Springer) (https://doi.org/10.1007/978-1-4614-7138-7)
[19] Aimé Safiah A-N, Yusoff M and Zeehaida M 2013 Colour image segmentation approach for detection of malaria parasites using various colour models and k-means clustering *WSEAS Transaction on Biology and Biomedicine* 10 41–52
[20] Pedregosa F et al 2011 Scikit-learn: machine learning in python *J. Mach. Learn. Res.* 12 2825–30
[21] Zheng X, Lei Q, Yao R, Gong Y and Yin Q 2018 Image segmentation based on adaptive K-means algorithm *EURASIP Journal on Image and Video Processing* 68
[22] Burney S A and Tariq H 2014 K-means cluster analysis for image segmentation *International Journal of Computer Applications* 96
[23] Dhachandira N, Manglem K and Chanu Y J 2015 Image segmentation using K-means clustering algorithm and subtractive clustering algorithm *Procedia Computer Science* 54 764–71
[24] Ng H P, Ong S H, Foong K W C, Goh P S and Nowinski W L 2006 *IEEE Southwest Symp. on Image Analysis and Interpretation*. 61–65
[25] Bagni T, Bovone G, Rack A, Mauro D, Barth C, Matera D, Buta F and Senatore C 2021 Machine learning applied to X-ray tomography as a new tool to analyze the voids in RRP Nb3Sn wires *Sci Rep.* 11 7767