Counting Molecules: Python based scheme for automated enumeration and categorization of molecules in scanning tunneling microscopy images

Jack Hellerstedt\textsuperscript{a,b}, Aleš Cahlík\textsuperscript{a,b}, Martin Švec\textsuperscript{a,b}, Oleksandr Stetsovych\textsuperscript{a}, Tyler Hennen\textsuperscript{c}

\textsuperscript{a}Institute of Physics of the Czech Academy of Sciences, v.v.i., Cukrovarnická 10, 162 00 Praha 6, Czech Republic
\textsuperscript{b}Regional Centre of Advanced Technologies and Materials, Palacký University, Šlechtitelů 27, 78371 Olomouc, Czech Republic
\textsuperscript{c}Institut für Werkstoffe der Elektrotechnik II, RWTH Aachen University, Sommerfeldstraße 24, 52056 Aachen, Germany

Abstract

Scanning tunneling and atomic force microscopies (STM/nc-AFM) are rapidly progressing to offer unprecedented spatial resolution of a diverse array of chemical species. In particular, they are employed to characterize on-surface chemical reactions by directly examining precursors and products. Chiral effects and self-assembled structures can also be investigated. This open source, modular, python based scheme automates the categorization of a variety of molecules present in medium sized (10\times 10 to 100\times 100 nm) scanned probe images.

Keywords: scanning tunneling microscopy, python, molecules, counting

1. Motivation and significance

Scanned probe techniques such as scanning tunneling microscopy (STM) and non-contact atomic force microscopy (nc-AFM) have now made direct investigation of on-surface reactions a routine experimental technique \[1,2\]. As the diverse ecosystem of organic molecules accessible by these techniques continues to grow so does the need for more sophisticated tools to extract quantitative information from larger and more complicated datasets of molecule imaging.

Email address: hellerstedt.jack@gmail.com (Jack Hellerstedt)

Preprint submitted to ArXiv March 7, 2022
Extracting statistics from STM images is often done by hand [3, 4, 5, 6]. This necessarily limits the size and complexity of statistical problems that can be tackled using these datasets. To address this limitation in extracting and categorizing molecules in STM images, we developed an automated scheme utilizing existing image processing libraries written using the Python programming language.

The existing widely used programs for the analysis of scanned probe data (WSxM [7] and Gwyddion [8]) do not presently incorporate tools for automated image processing and feature extraction. ImageJ [9] is a powerful and widely used piece of software in the biology community, but lacks compatibility and relevance for the datasets this package is designed to address. Another hindrance is that these image processing tools for biology applications are often written using commercial, closed source MATLAB code. Digital Surf has feature categorization tools but is also commercial and closed source [10].

There has been recent progress in developing machine-learning based tools for the automated assignment of atomic structure and defect information from scanned probe datasets, [11, 12, 13] as well as automated routines for improving the probe condition in STM [14, 15]. However these solutions address comparatively homogeneous datasets of more simple structures compared the present case, or require the computational resources and data (real or artificial) to train a convolutional neural network [15, 13]. We wanted to avoid this approach and instead develop a lightweight tool to quickly identify statistical trends in a large number of STM images with a diverse manifold of initially unknown molecular species; the functions offered in this package were used to prepare Fig. 3 in Hellerstedt et. al [16].

Our approach addresses some specific obstacles. There are practical limitations to the data resolution versus the number of species sampled. The sorting method needs account for different adsorption configurations of the same species (e.g., rotations and chirality). While explicit pairwise template matching has advantages in robustness, it suffers from the computational demands scaling in quadrature with the number of molecules to be sorted.

Our solution to these problems relies on the Zernike polynomial basis set to provide a “fingerprint” of coefficients representing every molecule [17, 18]. These coefficients are robust in response to rotations and noise in the data. In conjunction with other physically motivated coefficients, they provide the input to clustering algorithms for sorting the molecules. There are many different algorithms that can be applied to this type of data for the purposes of grouping like kinds of molecules together. We discuss some of the approaches we applied and their relative efficacy.
2. Software description

This package consists of a set of functions written using the Python libraries Numpy [19], scikit-image [20], scikit-learn [21], Mahotas [18] and Matplotlib [22]. In practice, it is a series of functions called sequentially as shown in Fig. 1. We developed the current implementation to be useful ‘out-of-the-box’ with only minimum skill prerequisites required (e.g., installing a python distribution and necessary packages). The modular nature of the data processing flow allows for a great deal of flexibility to address different datasets. The open source, repository based distribution of this package allows for further customization, and the possibility for community based development and improvement on these core functionalities.

2.1. Software Architecture

We first read in the image data and apply filters to make the image suitable for molecule template extraction using an adaptive threshold. After identifying closed contours corresponding to the perimeters of each molecule, we use the interior data of each contour as a representative template image. We then extract numerical features from each molecule, consisting of the calculated Zernike moments [17, 18] of the template images as well as the maximum topographic heights and the contour perimeter lengths. Finally, we use clustering algorithms [21] to categorize these features. Further functionality is provided to visualize the sorted categories, and make manual corrections.

Figure 1 shows the typical use of the provided functions. We rely on the Nanonispy library to read in data from one of the prevalent formats (Nanonis SXM files). This generates an array of pixel data and the rescaling factors to convert to real-space distances.

2.2. Software Functionalities

The default filtering function provided performs a Gaussian filter and plane fit subtraction of the image data. We calculate the global Otsu threshold [23] and use that to scale the offset value for feature extraction using a local thresholding method. We include diagnostic functions that can plot the filtered image data with the extracted contours, labeled by number, as well as a grid view of all the extracted molecule templates.

Zernike moments are the coefficients for representing an image decomposed into the orthogonal Zernike polynomial basis set [17]. This method has attracted attention and effort in the computer vision community for being rotationally invariant. This property is particularly useful in the context of our application because we wish to match molecules regardless of how
they are absorbed on the sample surface. We utilize the Mahotas library for calculating the Zernike moments for each molecule template [18]. We set the median template diagonal length as the default value of the Zernike radius input, an assumption that could fail depending on the homogeneity of the templates.

In addition to rotational invariance, these moments are insensitive to translation, mirroring, and rescaling. To account for differences in the real space footprint of each molecule, we additionally calculate the length, as well as the maximum pixel value within each contour.

With these characteristic moments and physical length scales, we perform a clustering analysis using the algorithms available in the Scikit-learn library [21]. For the datasets studied in this work, we found the Birch algorithm [24] with a threshold factor between 0.1 to 0.4 to be most effective at sorting images with no a priori knowledge of the number of categories. When the number of sorting categories is known, hierarchical clustering was found to be more accurate. The most effective sorting was accomplished using affinity propagation, where the cluster center preferences were defined using a hand-selected set of exemplar molecules.

The invariance of the Zernike moments to mirror symmetry in particular, makes them insensitive to differentiating between chiral molecules. Absolute quantifications of chirality are surprisingly difficult to define [25]. We developed a function to do a pairwise comparison of molecules within each sorted
category. By comparing each molecule and its mirror image, they can be sorted into right- and left-handed categories.

All these image, contour, correlations, and categorization data can be exported/saved for later use. Also included is a function using the interactive features of Matplotlib to manually categorize all the molecules.

3. Illustrative Examples

Representative outputs for the three example datasets from hand-selected exemplar molecules are shown in Figs. 2, 3, and 4. The work on the 9-azidophenanthrene system which was the source of the data in Figs. 2 and 3 provided the original impetus to develop this package [16]. The helicene data shown in Fig. 4 has been previously described in the literature [5]. These data were acquired via scanning tunneling microscopy at low temperature (∼5 K), with a pixel density of 10 nm⁻¹ to 13 nm⁻¹.

4. Expected Impact

This is the first effort to our knowledge to create an open source scheme to automate the task of counting and sorting hundreds of molecules, specifically tailored for scanning probe microscopy datasets. We expect this package to be immediately useful for anyone tasked with extracting population statistics from images comparable to those included as examples. This tool has
Figure 3: 9-azidophenanthrene molecules on Ag(111) sorted into 9 groups using hand-selected exemplars. 50×50 nm, 512×512 pixels.

Figure 4: Helicene data sorted using hand-selected exemplars. The dimers (categories 1 and 4) were subsequently sorted by chirality. 80×80 nm, 1024×1024 pixels. Compare to Fig. 4 of Stetsovych et. al. [5].
already proven useful by making it easy to quickly extract and visualize population statistics in image datasets. By scaling the statistics and quantitative information that can be extracted from these types of datasets, we anticipate that this will facilitate the ability to answer novel and more nuanced questions about how chemical processes unfold on a surface. We hope to reveal and address limitations by having this set of functions applied to the increasingly diverse ecosystem of on-surface chemistry being explored by the greater scanning probe community.

5. Conclusions

Here we have presented our efforts to automate the counting and sorting of molecules in topographic images acquired with a scanning tunneling microscope. The example datasets, images with several hundred molecules, can be sorted on a personal computer in seconds. The python based, open source, modular design of the components strives to be immediately accessible to a non-expert user, and allow for significant modification/ customization to suit individual needs. We hope that use and feedback from the wider community will allow this tool to continue to be developed and provide utility for advancing the understanding of on-surface chemistry experiments.

6. Software/ Data availability

This package is available on GitHub and archived at: https://doi.org/10.5281/zenodo.6324850
The data in Figures 2, 3, 4 is available on figshare: https://doi.org/10.6084/m9.figshare.19217556

Acknowledgements

S. Edalatmanesh and A. Caparrós provided useful discussion in developing the solution to the chirality sorting problem. We thank Dr. Karthäuser for critical reading of this manuscript. J.H. acknowledges the support of the CIES/ Czech Fulbright commission during the development of this package.

References

[1] J. V. Barth, Molecular Architectonic on Metal Surfaces, Annual Review of Physical Chemistry 58 (1) (2007) 375–407. doi:10.1146/annurev.physchem.56.092503.141259
URL http://www.annualreviews.org/doi/abs/10.1146/annurev.physchem.56.092503.141259
[2] P. Jelínek, High resolution SPM imaging of organic molecules with functionalized tips, Journal of Physics: Condensed Matter 29 (34) (2017) 343002. doi:10.1088/1361-648X/aa76c7 URL http://stacks.iop.org/0953-8984/29/i=34/a=343002?key=crossref.eee0848de87e9b5c6b61380d39a7cee4

[3] M. C. Capsoni, On-surface self-assembly and characterization of a macromolecular charge transfer complex by scanning tunneling microscopy and spectroscopy, Ph.D. thesis, University of British Columbia (2016). doi:10.14288/1.0305019 URL https://open.library.ubc.ca/cIRcle/collections/ubctheses/24/items/1.0305019

[4] J. Prinz, O. Gröning, H. Brune, R. Widmer, Highly enantioselective adsorption of small prochiral molecules on a chiral intermetallic compound, Angewandte Chemie - International Edition 54 (13) (2015) 3902–3906. doi:10.1002/anie.201410107

[5] O. Stetsovych, M. Švec, J. Vacek, J. V. Chocholoušová, A. Jančařík, J. Rybáček, K. Kosmider, I. G. Stará, P. Jelínek, I. Starý, From helical to planar chirality by on-surface chemistry, Nature Chemistry 9 (3) (2016) 213–218. doi:10.1038/nchem.2662 URL http://www.nature.com/doifinder/10.1038/nchem.2662

[6] F. D. Goll, G. Taubmann, U. Ziener, Static Scanning Tunneling Microscopy Images Reveal the Mechanism of Supramolecular Polymerization of an Oligopyridine on Graphite, Angewandte Chemie International Edition doi:10.1002/anie.202117580 URL https://onlinelibrary.wiley.com/doi/10.1002/anie.202117580

[7] I. Horcas, R. Fernandez, J. M. Gomez-Rodriguez, J. Colchero, J. Gomez-Herrero, A. M. Baro, WSXM: A software for scanning probe microscopy and a tool for nanotechnology, Review of Scientific Instruments 78 (1) (2007) 013705. doi:10.1063/1.2432410 URL http://scitation.aip.org/content/aip/journal/rsi/78/1/10.1063/1.2432410

[8] D. Nečas, P. Klapetek, Gwyddion: An open-source software for SPM data analysis, Central European Journal of Physics 10 (1) (2012) 181–188. doi:10.2478/s11534-011-0096-2
[9] C. T. Rueden, J. Schindelin, M. C. Hiner, B. E. DeZonia, A. E. Walter, E. T. Arena, K. W. Eliceiri, ImageJ2: ImageJ for the next generation of scientific image data, BMC Bioinformatics 18 (1) (2017) 1–26. doi:10.1186/s12859-017-1934-z.

[10] M. Cognard, Digital Surf (2020). URL https://perma.cc/4K5V-JQREhttps://www.digitalsurf.com/news/perform-a-particle-analysis-on-microscopy-images/

[11] M. Scherbela, L. Hörmann, A. Jeindl, V. Obersteiner, O. T. Hofmann, Charting the energy landscape of metal/organic interfaces via machine learning, Physical Review Materials 2 (4) (2018) 043803. doi:10.1103/PhysRevMaterials.2.043803.
URL http://arxiv.org/abs/1709.05417https://link.aps.org/doi/10.1103/PhysRevMaterials.2.043803

[12] M. Ziatdinov, O. Dyck, A. Maksov, X. Li, X. Sang, K. Xiao, R. R. Unocic, R. Vasudevan, S. Jesse, S. V. Kalinin, Deep Learning of Atomically Resolved Scanning Transmission Electron Microscopy Images: Chemical Identification and Tracking Local Transformations, ACS Nano 11 (12) (2017) 12742–12752. doi:10.1021/acsnano.7b07504.
URL http://pubs.acs.org/doi/10.1021/acsnano.7b07504

[13] J. Li, M. Telychko, J. Yin, Y. Zhu, G. Li, S. Song, H. Yang, J. Li, J. Wu, J. Lu, X. Wang, Machine Vision Automated Chiral Molecule Detection and Classification in Molecular Imaging, Journal of the American Chemical Society 143 (27) (2021) 10177–10188. doi:10.1021/jacs.1c03091.
URL https://pubs.acs.org/doi/10.1021/jacs.1c03091

[14] M. Rashidi, R. A. Wolkow, Autonomous Scanning Probe Microscopy in Situ Tip Conditioning through Machine Learning, ACS Nano (2018) acsnano.8b02208doi:10.1021/acsnano.8b02208.
URL http://arxiv.org/abs/1803.07059http://pubs.acs.org/doi/10.1021/acsnano.8b02208

[15] A. Krull, P. Hirsch, C. Rother, A. Schiffrin, C. Krull, Artificial-intelligence-driven scanning probe microscopy, Communications Physics 3 (1) (2020) 54. doi:10.1038/s42005-020-0317-3.
URL http://dx.doi.org/10.1038/s42005-020-0317-3http://www.nature.com/articles/s42005-020-0317-3

[16] J. Hellerstedt, A. Cahlík, O. Stetsovych, M. Švec, T. K. Shimizu, P. Mutombo, J. Klívar, I. G. Stará, P. Jelínek, I. Starý, Aromatic Azide
Transformation on the Ag(111) Surface Studied by Scanning Probe Microscopy, Angewandte Chemie International Edition 58 (8) (2019) 2266–2271. doi:10.1002/anie.201812334
URL http://doi.wiley.com/10.1002/anie.201812334

[17] A. Khotanzad, Y. H. Hong, Invariant Image Recognition by Zernike Moments, Ann. Oper. Res. Pattern Anal. Machine Intell. IEEE Trans. Pattern Anal. Machine Intell. J. Robotics Res. J. Robotics Res. J. ACM Networks I. J. Stoker 12 (14) (1990) 13–118. doi:10.1109/34.55109

[18] L. P. Coelho, Mahotas: Open source software for scriptable computer vision, Journal of Open Research Software 1 (1) (2013) e3. doi:10.5334/jors.ac. URL http://openresearchsoftware.metajnl.com/articles/10.5334/jors.ac/

[19] T. E. Oliphant, Python for Scientific Computing, Computing in Science & Engineering 9 (3) (2007) 10–20. doi:10.1109/MCSE.2007.58. URL http://ieeexplore.ieee.org/document/4160250/

[20] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, scikit-image: image processing in Python, PeerJ 2 (2014) e453. doi:10.7717/peerj.453. URL https://peerj.com/articles/453

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[22] J. D. Hunter, Matplotlib: A 2D Graphics Environment, Computing in Science & Engineering 9 (3) (2007) 90–95. doi:10.1109/MCSE.2007.55. URL http://ieeexplore.ieee.org/document/4160265/

[23] N. Otsu, A Threshold Selection Method from Gray-Level Histograms, IEEE Transactions on Systems, Man, and Cybernetics 9 (1) (1979) 62–66. doi:10.1109/TSMC.1979.4310076. URL http://ieeexplore.ieee.org/document/4310076/

[24] T. Zhang, R. Ramakrishnan, M. Livny, BIRCH: an efficient data clustering method for very large databases, ACM SIGMOD Record 25 (2) (1996) 103–114. doi:10.1145/235968.233324. URL http://portal.acm.org/citation.cfm?doid=235968.233324
[25] A. B. Buda, T. A. der Heyde, K. Mislow, On Quantifying Chirality, Angewandte Chemie International Edition in English 31 (8) (1992) 989–1007. doi:10.1002/anie.199209891
URL http://doi.wiley.com/10.1002/anie.199209891