The Journal of Open Source Software

Bramble: adaptive common neighbor analysis (CNA) for the recognition of surface topologies in nanoparticles

I. A. W. Filot [1¶], **M. P. C. van Etten** [1], **D. W. J. G. Trommelen** [1], and **E. J. M. Hensen** [1]

1 Inorganic Materials and Catalysis, Department of Chemical Engineering and Chemistry, Eindhoven University of Technology ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

In heterogeneous catalysis, the active site is the specific region on the catalyst surface where the chemical reaction occurs. It plays a vital role in facilitating reactions by providing a unique arrangement of atoms and chemical properties to promote the conversion of reactants into products. Identifying these atomic motifs is crucial towards the atom-scale understanding of catalysis. Towards this aim, the common neighbor analysis (CNA) procedure acts as a powerful computational method used to analyze and classify atomic structures in materials. It involves examining the local environment of each atom to identify and quantify the types of atomic coordination and bonding, providing insights into the structural properties and behavior of materials at the atomic scale. `Bramble` is an efficient C++-based command-line tool to perform the CNA analysis. Uniquely, it is coupled to a pattern identification library for facile identification of the CNA fingerprints. Furthermore, for unknown fingerprint it offers the option to perform a similarity analysis based on the minimization of the Hilbert-Schmidt norm.

Statement of need

In heterogeneous catalysis, the active site refers to the specific location on the catalyst surface where the chemical reaction takes place. It is a region characterized by its unique arrangement of atoms and chemical properties that facilitate the conversion of reactants into products. The active site provides an environment that can selectively adsorb reactant molecules, weaken chemical bonds, and promote reaction pathways. Understanding and optimizing the active site is crucial for designing efficient catalysts and improving catalytic processes in diverse applications such as industrial chemical production and environmental remediation.(Santen, 2017)

Understanding of the catalytic activity of nanoparticles is complex due to the large array of potentially important active site configurations present. To automate the procedure of recognizing these surface motifs, the Common Neighbor Analysis (CNA) method offers a powerful computational approach used to enumerate the different atomic structures present at the local scale. (Faken & Jónsson, 1994) It identifies coordination, classifies bonding, and provides quantitative descriptors. The CNA method aids in understanding structural properties, material behavior, and designing materials for specific applications. It plays a crucial role in unraveling atomic structures and bonding patterns, enabling tailored material development.(Faken & Jónsson, 1994; Stukowski, 2012)

There exist a variety of programs and tools that support the CNA procedure. Perhaps the most well-known are Ovito(Stukowski, 2010) and Pyscal (Menon et al., 2019). Ovito implements a

rich set of various CNA flavours, including conventional CNA, adaptive CNA, interval CNA and bond-based CNA. Pyscal is an extensive python module for structural analysis of atomic environment using an efficient C++-based back-end. Both tools offer excellent documentation and are used throughout the field, however mainly focus on bulk structures and the identification of crystal phases. As such, they are fairly limited in the description of surface atoms and the recognition of active sites, which is the target of `Bramble`.

`Bramble` is a C++-based command-line tool for the evaluation of CNA fingerprints, based on the adaptive CNA flavor, with the aim of recognizing atomic configurations pertaining to active sites. Uniquely, the `Bramble` CNA tool is bundled alongside a pattern library allowing for the association of convenient and clear labels to the CNA fingerprints. When a given fingerprint is not recognized by the program, a similarity analysis can be performed to quantitatively assess how similar or different the local atomic environment is with respect to the other atoms in the systems. `Bramble` has already been used in a number of scientific publications (Etten et al., 2021; Sterk et al., 2022). `Bramble` uses a minimal set of dependencies, i.e. Boost(Koranne, 2011), `TCLAP`(Smoot et al., 2009), and Eigen3 (Guennebaud et al., 2010), which are all readily available on modern Linux based operating systems. Optionally, `Bramble` can make use of GPU acceleration via a CUDA-based(NVIDIA et al., 2020) acceleration module.

Internally, `Bramble` uses an adaptive CNA flavor largely based on the work of Reinhart and coworkers.(Reinhart et al., 2017) The nearest neighbors used for determining the local atomic environment are based on a cut-off distance r_{cut} as given by an weighted average distance of the distance to the six nearest neighbors

$$r_{\text{cut}} = \left(\frac{1+\sqrt{2}}{2}\right)\left(\frac{1}{6}\sum_{j=1}^{6}|\vec{r}_{ij}|\right).$$

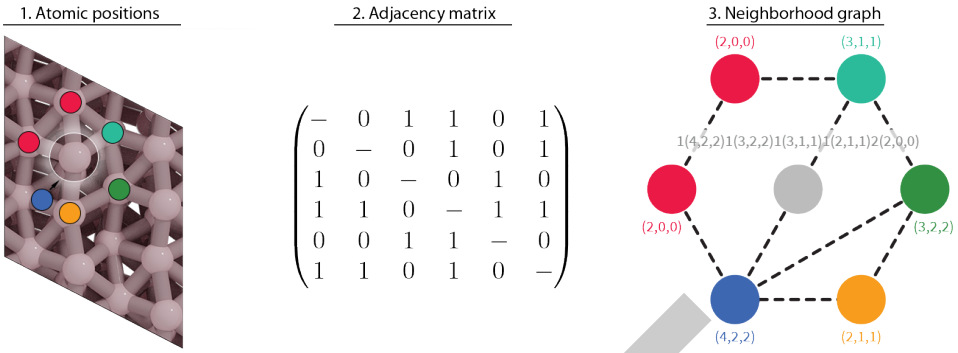
Using this cut-off distance, a boolean distance matrix is constructed between all the neighboring atoms that lie at a distance $r < r_{\text{cut}}$ with respect to the atom under investigation. In this boolean distance matrix, when the distance between any two neighboring atoms i and j is such that $r_{ij} < r_{\text{cut}}$, a 1 is placed in the matrix and a 0 otherwise. Essentially, this boolean distance matrix is a representation of the neighborhood graph. For each node in this neighborhood graph, a triplet of CNA indices are calculated corresponding to

1. The number of nearest neighbors.
2. The number of edges between connecting those nodes to each other.
3. The length of the longest path continuous path among those edges.

To construct the CNA fingerprint, the number of times a given triplet of CNA indices is encountered is recorded and bundled into a string representation. A schematic depiction of the CNA algorithm is given in Figure 1.

For a large number of stable surface terminations of face-centered cubic (FCC), hexagonal close-packed (HCP), body-centered cubic (BCC) and simple cubic (SC) type of crystals the CNA fingerprints have been established and collected in a convenient fingerprint library. When a CNA fingerprint is known, the corresponding label is offered by the program which helps the user in the identification of the surface atom. This pattern library can be easily extended using a command-line utility `brambletool`.

The Journal of Open Source Software



$$\begin{pmatrix} - & 0 & 1 & 1 & 0 & 1 \\ 0 & - & 0 & 1 & 0 & 1 \\ 1 & 0 & - & 0 & 1 & 0 \\ 1 & 1 & 0 & - & 1 & 1 \\ 0 & 0 & 1 & 1 & - & 0 \\ 1 & 1 & 0 & 1 & 0 & - \end{pmatrix}$$

Figure 1: Schematic depiction of the CNA algorithm. (1) Geometry of a Cobalt (11-21) surface termination. For the highlighted atom, the CNA fingerprint is going to be determined. Based on the cutoff distance, this atom has six neighbors as indicated by the circles. (2) For these neighbors, an adjacency matrix is constructed. For each off-diagonal element in this matrix a value of one is assigned if the distance of the corresponding two neighbors is less than the cutoff distance. If not, the element has a value of zero. (3) For each node in the adjacency matrix, a triplet of CNA indices is determined corresponding to (i) the number of nearest neighbor nodes, (ii) the number of edges connecting those nodes to each other, and (iii) the length of the longest continuous path among those edges..

When a given CNA fingerprint cannot be identified, i.e. when that fingerprint is unknown to the library, the user can perform a similarity analysis. For each atom in a given set, a metric is given how similar that atom is to each other atom in the same set. This allows the user to at least characterize the atom to how similar it is to a known atom. Given two distance matrices corresponding to the distance between each pair of neighboring atoms as given above, the Hilbert-Schmidt norm of the difference between these two matrices can be calculated as given by

$$\mu_{ij} = \min_{\mathbf{P}} ||\mathbf{P}\mathbf{D}(i)\mathbf{P}^{\dagger} - \mathbf{D}(j)||,$$

wherein $||\cdot||$ is the Hilbert-Schmidt norm as given by

$$||\mathbf{D}|| = \sum_{ij} D_{ij}^2$$

and $\mathbf{P}$ is a permutation matrix. Essentially, the minimization loops over all possible $N!$ permutation for a given $N \times N$ distance matrix. Because establishing the minimum distance is essentially tied to a graph isomorphism problem, a brute-force approach is used to ensure that the lowest μ_{ij} is found.

An extensive user guide including examples, compilation instructions and documentation of the command line arguments is available at https://bramble.imc-tue.nl/.

Acknowledgements

This work was supported by the Netherlands Center for Multiscale Catalytic Energy Conversion, and NWO Gravitation program funded by the Ministry of Education, Culture and Science of the government of the Netherlands. The Netherlands Organization for Scientific Research is acknowledged for providing access to computational resources. Sven Roefs, Koen Jongejan, and Rudie van den Berg are acknowledged for providing useful comments and suggestions for the validation of the algorithm and the generation of the dataset used in the construction of the pattern recognition library.

References

Etten, M. P. C. van, Zijlstra, B., Hensen, E. J. M., & Filot, I. A. W. (2021). Enumerating active sites on metal nanoparticles: Understanding the size dependence of cobalt particles for CO dissociation. *ACS Catalysis*, *11*(14), 8484–8492. https://doi.org/10.1021/acscatal.1c00651

Faken, D., & Jónsson, H. (1994). Systematic analysis of local atomic structure combined with 3D computer graphics. *Computational Materials Science*, *2*(2), 279–286. https://doi.org/10.1016/0927-0256(94)90109-0

Guennebaud, G., Jacob, B., & others. (2010). *Eigen v3*. http://eigen.tuxfamily.org.

Koranne, S. (2011). Boost c++ libraries. In *Handbook of open source tools* (pp. 127–143). Springer US. https://doi.org/10.1007/978-1-4419-7719-9_6

Menon, S., Leines, G. D., & Rogal, J. (2019). Pyscal: A python module for structural analysis of atomic environments. *Journal of Open Source Software*, *4*(43), 1824. https://doi.org/10.21105/joss.01824

NVIDIA, Vingelmann, P., & Fitzek, F. H. P. (2020). *CUDA, release: 10.2.89*. https://developer.nvidia.com/cuda-toolkit

Reinhart, W. F., Long, A. W., Howard, M. P., Ferguson, A. L., & Panagiotopoulos, A. Z. (2017). Machine learning for autonomous crystal structure identification. *Soft Matter*, *13*, 4733–4745. https://doi.org/10.1039/C7SM00957G

Santen, R. A. van. (2017). Heterogeneous catalysis. In *Modern heterogeneous catalysis*. John Wiley & Sons, Ltd. https://doi.org/10.1002/9783527810253.ch1

Smoot, M. E., Aarno, D., & others. (2009). *Templatized c++ command line parser library*. https://tclap.sourceforge.net/.

Sterk, E. B., Nieuwelink, A.-E., Monai, M., Louwen, J. N., Vogt, E. T. C., Filot, I. A. W., & Weckhuysen, B. M. (2022). Structure sensitivity of CO2 conversion over nickel metal nanoparticles explained by micro-kinetics simulations. *JACS Au*, *2*(12), 2714–2730. https://doi.org/10.1021/jacsau.2c00430

Stukowski, A. (2010). Visualization and analysis of atomistic simulation data with OVITO-the open visualization tool. *Modelling and Simulation in Materials Science and Engineering*, *18*(1). https://doi.org/10.1088/0965-0393/18/1/015012

Stukowski, A. (2012). Structure identification methods for atomistic simulations of crystalline materials. *Modelling and Simulation in Materials Science and Engineering*, *20*(4), 045021. https://doi.org/10.1088/0965-0393/20/4/045021