HHS Public Access

Author manuscript

J Chem Inf Model. Author manuscript; available in PMC 2022 November 10.

Published in final edited form as:

J Chem Inf Model. 2020 December 28; 60(12): 6251–6257. doi:10.1021/acs.jcim.0c00899.

Chespa: streamlining expansive chemical space evaluation of molecular sets

Jamie R. Nuñez[1,2], **Monee Mcgrady**[1], **Yasemin Yesiltepe**[1,2], **Ryan S. Renslow**[1,2,*], **Thomas O. Metz**[1,†]

[1]Earth and Biological Sciences Directorate, Pacific Northwest National Laboratory, Richland, WA, USA 99352

[2]The Gene and Linda Voiland School of Chemical Engineering and Bioengineering, Washington State University, Pullman, WA, USA 99164

Abstract

Thousands of chemical properties can be calculated for small molecules, which can be used to place the molecules within the context of a broader "chemical space." These definitions vary based on compounds of interest and the goals for the given chemical space definition. Here, we introduce a customizable (i.e., modular) Python module, *chespa*, built to easily assess different chemical space definitions through clustering of compounds in these spaces and visualizing trends of these clusters. To demonstrate this, *chespa* currently streamlines prediction of various molecular descriptors (predicted chemical properties, molecular substructures, AI-based chemical space, and chemical class ontology) in order to test 6 different chemical space definitions. Furthermore, we investigated how these varying definitions trend with mass spectrometry (MS)-based observability, i.e., the ability of a molecule to be observed with MS (e.g., as a function of the molecule ionizability), using an example data set from the U.S. EPA's Non-Targeted Analysis Collaborative Trial (ENTACT), where blinded samples had been analyzed previously, providing 1,398 data points. Improved understanding of observability would offer many advantages in small molecule identification, such as (i) *a priori* selection of experimental conditions based on suspected sample composition, (ii) the ability to reduce the number of candidate structures during compound identification by removing those less likely to ionize, and, in turn, (iii) a reduced false discovery rate and increased confidence in identifications. Factors controlling observability are not fully understood, making prediction of this property non-trivial and a prime candidate for chemical space analysis. *Chespa* is available at github.com/pnnl/chespa.

Graphical Abstract

*Corresponding Author: Ryan S. Renslow - ryan.renslow@pnnl.gov. †Co-corresponding Author: Thomas O. Metz - thomas.metz@pnnl.gov.

Author Contributions. The manuscript was written through contributions of all authors. All authors have given approval to the final version of the manuscript.

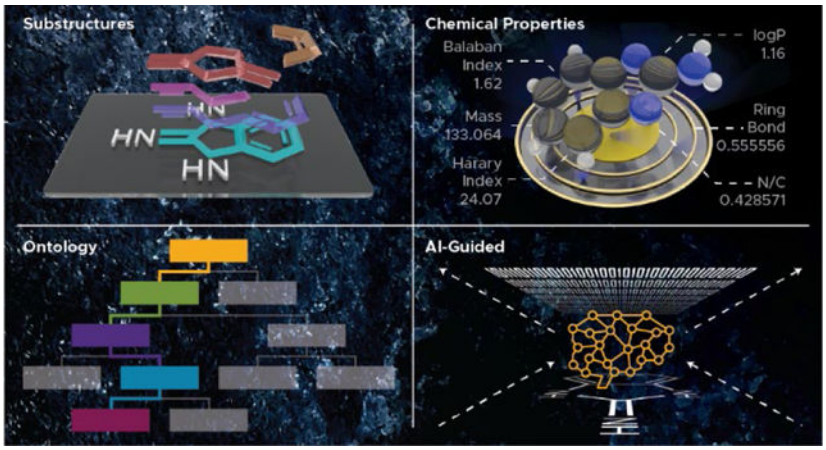
ASSOCIATED CONTENT

chespa. Available at github.com/pnnl/chespa.

Supporting Information. The Supporting Information is available free of charge on the ACS Publications website.

SupportingInformation.pdf: Includes further detailed methods and additional figures.

SupportingData.xlsx: Includes our suspect library, property predictions, and results. Table captions are provided in SupportingInformation.pdf



INTRODUCTION

The discipline of metabolomics seeks to confidently identify and quantify (i.e. measure) the entire small molecule composition of a given sample. The composition and complexity of a metabolome can be described by placing the representative compounds within the context of a broader “chemical space.” Chemical space is the multidimensional space defined by a set of descriptors, in which each dimension is represented by either empirical or computational descriptors, and compounds are placed in this space based on their property values compared to all other molecule property values.[1] Thousands of molecular properties can be calculated for a single compound using the wide range of cheminformatics, machine learning, and quantum chemistry-based software currently available.[2–4] Although historically scientists have tended toward chemical property-based chemical spaces, the descriptors that define a chemical space do not have to be solely based on chemical properties. A chemical space can be built using principal components from a principal component analysis (derived from many sources, including traditional properties), latent vectors from artificial intelligence-based latent space, ontological categorization of the compounds, substructure enumeration, and others. Due to the nature of how chemical space is defined, the size of the space depends on how many descriptors are chosen, how many molecules are used to define that space, and other limitations placed on it. For example, it is estimated that there are over 10^{60} compounds in druglike chemical space, which is only one example of a limited chemical space (e.g., in this case, small molecules made up of only carbon, nitrogen, oxygen, and sulfur atoms[5]), yet other chemical spaces are estimated to expand beyond 10^{100} compounds.[6]

These spaces are useful for researchers in many ways, including to explore the metabolome,[7–8] search the space of lead-like compounds for potential drugs,[9–10] generate new compounds to fill gaps in the space,[11–13] and predict properties of molecules.[14–16] There are tools available that can easily visualize various definitions of chemical space,[17] generate novel compounds,[11, 18] and predict their associated properties,[18] but there are still many gaps in the field of chemical space analysis. Most chemical spaces created to date are focused on a small set of molecules in order to find new structures with similar properties,[19–20] and so larger chemical spaces have not yet been fully explored. It is

important to analyze these larger chemical spaces as there are unpopulated areas within them that could contain valuable chemical information,[21] although, to date, there have not been many tools to automate chemical space analysis. Tools that have been made available to date are not open-access, open-source, nor built to assess any given chemical space.[22]

Toward this end, we introduce a Python module, *chespa*, that can be used to assess compounds based on varying definitions of chemical space. Specifically, *chespa* provides an adjustable framework for expansive chemical space analysis, which in this first instantiation provides automation of chemical space based on predicted chemical properties, molecular substructures, AI-based chemical space, and chemical class ontology. For each of these different chemical space aspects, as an initial implementation, we based our chemical property-derived chemical spaces on principal component analysis (PCA) of properties from ChemAxon, three substructure-based chemical spaces (one based on Open Babel's[23] MACCS and two from SPECTRe[24]), our AI-derived chemical space on a recent variational auto-encoder, DarkChem,[18] and our initial chemical class ontology on ClassyFire.[25] However, *chespa* was built so other chemical spaces, properties, ontologies, and AI-based methods could be easily incorporated in the future. This module was designed to automate clustering of compounds based on where they fall across these varying chemical space definitions, assess how each definition of chemical space performs, and use this information to investigate trends in measurable chemical properties.

One such property that can be investigated using chemical space is mass spectrometry (MS)-based observability: the ability to use MS to detect a compound given some set of experimental conditions. The observability of a molecule using MS is highly variable as it can vary according to the innate ionizability of a compound, MS instrument sensitivity, sensitivity of connected instrumentations (e.g. ion mobility spectrometry; IMS), matrix effects (e.g. influences of concentration, other compounds present), chromatography techniques (e.g. mobile phase composition, flow rate), ionization source parameters (e.g. temperature, voltage) and general treatment of the output data (post-processing and adducts/multimers searched for).

Here, we show an example use of *chespa*: analysis of molecule observability in the set of blinded complex mixtures created for the U.S. Environmental Protection Agency's (EPA) Non-Targeted Analysis Collaborative Trial (ENTACT),[26] a study focused on testing current metabolite identification techniques. Previously,[27] we studied these samples and provided evidence of presence for compounds in the samples based, in part, on *m/z* and collision cross section (CCS) values collected using electrospray ionization (ESI) coupled with IMS-MS. In this paper, we use the information from 545 true positives and 853 true negatives (as confirmed by the EPA after unblinding) and present our findings on how our instrumentation performed with these samples when focusing on protonated, deprotonated, and sodiated adducts.

Our goal here is to describe *chespa*, demonstrate its application for a simple dataset, and lay the foundation for rigorous assessment of various properties in order to better understand observability. We provide *chespa* as an open source Python module, and all code and data reported for this analysis is available at github.com/pnnl/chespa, complete with a Binder[28]

(an open-source sharable, executable environment available online) and working data for easy testing and exploration.

METHODS

chespa.

Chespa is a Python module made to streamline calculation of chemical descriptors and chemical spaces, perform and assess clustering of molecular sets in chemical spaces, compare differences between chemical spaces, and generate plots to evaluate and visualize the results. *Chespa* combines the functionality of several disparate pre-existing and custom in-house tools, in order to facilitate chemical space analysis of large sets of molecules. *Chespa* consists of 4 scripts that assist in the prediction of chemical descriptors described here (i.e. wrappers for DarkChem, ClassyFire, ChemAxon, and SPECTRe, though it should be noted these tools are not included as part of *chespa* and must be downloaded separately), 2 scripts that aid in chemical space data processing and plotting, and 2 interactive Jupyter Notebooks (v6.0.1) that assist in piecing together the analysis workflow. The architecture of *chespa* was designed so that new or custom tools can be easily added. Note, unlike the other tools used here, ChemAxon does require a license to access all functionality, but licenses are typically free for academic users. Alternatively, ChemAxon can be easily swapped out for a different property calculator. To lead the interested reader through each step of *chespa*-based analysis shown in this application note and associated Supplemental Information, a Binder[28] was created. This Binder enables full testing of *chespa* through the browser, and the link to launch it is available in the GitHub repo (github.com/pnnl/chespa). Because *chespa* is available online using this platform, their servers can be used for processing; however, to save any modifications made to this environment, a local downloaded copy is required. As can be seen in this Binder, Python 3.7.6 (and it's standard library) was used alongside numpy (v1.16.5),[29] scikit-learn (v0.21.3),[30] bmdcluster (v0.3.1),[31] and pandas (v0.25.1),[32] for data processing. RDKit (v2020.03.2),[33] seaborn (v0.9.0),[34] and matplotlib (v3.1.1)[35] were used for generating figures.

Timings for this tool and the calculators we use in this paper are available in Table S1. It should be noted that there is no inherent limit in the number of compounds or dimensions that can be run through chespa, but increasing the number of dimensions for any given chemical space will increase the time required for this tool to run, and should be considered during selection of chemical descriptors while also considering the number of compounds, the user's own computational resources, and allowable processing time.

Example Dataset.

Experimental data was collected from the analysis of 10 samples provided through the ENTACT study.[26] Full experimental details are provided in Nunez et al.[27] All samples were analyzed with an Agilent 6560 drift tube IMS coupled with a quadrupole time-of-flight mass spectrometer[36–37] using ESI in both positive and negative ionization modes. During this analysis, we labeled compounds as present or not present based on a scoring metric output by the Multi-Attribute Matching Engine (MAME). Here, we refer only to the 545 compounds that were "observed" in our analysis (i.e., true positives; meaning labeled as

present and confirmed by the EPA that they were spiked into the given sample), and 853 that were “not observed” (i.e., true negatives; meaning no evidence was found of their presence) by ESI after scoring optimization. Example data is shown in Figure S1.

The EPA’s ToxCast library was provided as a suspect library for the ENTACT challenge. For the analysis of samples described above, we processed compounds in this library to their expected most “ionizable” form (desalted, neutralized, major tautomer). This modified library, composed of 4,346 compounds, is used here and refered to as our suspect library.

Data Preparation.

Properties were added to the suspect library using *chespa*. ClassyFire,[25] a web-based, automated chemical classification tool, was used to define the superclass of each compound. DarkChem[18] latent space vectors were also added as LS1-128 (representing the 128 dimensions used to define its latent space). Helper functions are included in *chespa* to facilitate the calculation of these properties (helper_darkchem.py and helper_classyfire.py).

A chemical space (here, called the Property Chemical Space; PCS) was defined previously,[18] and was built from ~91 million compounds and 10 chemical properties. To fit the ENTACT suspect library into this space, the same properties were calculated for these compounds. Five were calculated with ChemAxon’s *cxcalc* (v18.8.0): ring bond percent, pKa of the most acidic atom, logP, Harary index, and Balaban index. The remaining 5 were exact (monoisotopic) mass and atom ratios (N/H, N/C, O/H, and O/C). These ratios were calculated using formula-processing code (formula_module.py) and molmass.py/elements.py (which were provided by Christoph Gohlke at the University of California, Irvine). All of this code is made available as part of *chespa*. Once these properties were calculated, compounds were placed into PCS using PCS’s chemical property averages and standard deviations and principal component analysis (PCA) results. First, mean imputation was performed using the averages calculated for the PCS. Then, data for each property was normalized by dividing by the standard deviation of its respective variable in the PCS and subtracting the average of its respective variable in the PCS. Principal components from the PCS were then used to transform this data and calculate the 10 principle component variables, PC1-10. See Supporting Data for all calculated values and the code provided with this paper (specifically, helper_chemspace.py for streamlined prediction of these values). All PCS variables described here are also available in the repository shared with this paper (located in data/pca).

Substructures were found using Open Babel’s MACCS fingerprints (166 pre-defined substructures).[23] These were then converted into a binary matrix, where each row represents a compound and each column represents a substructure. Cells are filled with a 1 if the given substructure was found in that compound, and a 0 if not. Empty columns (meaning no compound in the suspect library contained that given substructure) were removed.

Substructures were also found using SPECTRe, a Python tool that applies the concept of subgraph isomorphism in chemical search to find all substructures in a given compound, regardless of the substructure size.[24] A helper function is provided to aid in the generation of these substructures (helper_substructures.py). Data here is similar to that used for MACCS

(a binary matrix), except the number of substructures is not pre-defined, all substructures found for the suspect library are used.

Clustering.

Scikit-learn's (v.0.16.1)[38] KMeans class, provided in the *chespa* cluster module, was used for clustering Property Chemical Space PC vectors (PC1-10) and DarkChem latent space vectors (LS1-128). Default parameters were used except the random_state was set to 10. Silhouette analysis, also available as part of Scikit-learn, was used to determine the appropriate number of clusters (n_clusters).[39] For substructures, clustering was performed using Python package bmdcluster,[31] due to this data being binary.

RESULTS AND DISCUSSION

Grouping and Clustering.

Property chemical space clusters (referred to here as ChemSpace clusters) were produced by first fitting compounds from the suspect library into the chemical space defined[18] then clustered using KMeans. The chemical space covered by compounds from the suspect library, and the first 2 principal components (covering 51% variance), is shown in Figure 1. Silhouette analysis was performed and 8 clusters was deemed the best balance between the number of groups and group sizes (Figure S3). A complete breakdown of the number of compounds in each of these clusters (and following groups/clusters), given the full suspect library or only spiked in, observed, or not observed compounds, is shown in Figure S4. Statistics on the distribution and sizes of these clusters (and following discussed in this section) are available in the Supporting Information (SI RD1).

For the DarkChem clusters, silhouette analysis was performed again (Figure S5) and 8 clusters was deemed the best balance between the number of groups and group sizes.

In the case of ClassyFire, due to being a single categorical column, the top 7 most commonly found superclasses in the suspect library were labeled ClassyFire Superclass groups 1–7. Compounds that fell into any other superclass, or that were left unlabeled (ClassyFire is based on classifications used in the literature in the past, so not all compounds receive a label at all levels of the provided hierarchy), were placed into an eighth group. This total number of groups was desirable due to the results from Property Chemical Space and DarkChem.

Since SPECTRe is not limited by substructure size, or a pre-defined list, 5,502,426 substructures were found for this suspect library. To reduce sparse data, substructures that were represented in less than 1% of the suspect library were removed, leaving 1,288 total substructures. Again, due to the results from Chemical Space and DarkChem Substructure clusters, 8 clusters were used when producing clusters based on MACCS and SPECTRe substructures.

As a quick look into the molecular structures each of these groups/clusters are composed of, 5 randomly chosen representative compounds falling into each of them is shown in Figure 2 and Figure S6-S10.

Analysis of Trends in Observability.

The distribution of compounds in each of these groups/clusters was compared to find potential trends for molecules that could and could not be observed (Figure 3). Relative difference and the number of members in each cluster were also used to investigate differences between the amount of compounds observed or not for each cluster (Figure S11). To show how the distributions of compounds in each cluster look in respect to only those compounds observed, a similar figure was created using the number of total observed compounds (rather than spiked in) as the denominator for compounds observed with (+)−ESI and (−)−ESI (Figure S12). Combined results for compounds observed by either (+)−ESI or (−)−ESI (i.e., a view into how ESI performed in general) are also shown in Figure S13. Analysis was also done for the 6 maximum substructures, found using SPECTRe (Figure S14), but due to extensive overlap between groups and insignificant trends, these results were not investigated further.

To enable the investigation of how chemical properties may affect observability, the average properties (using the 10 variables in the definition of Property Chemical Space) of compounds in each of the groups/clusters for the full library, spiked in compounds, or compounds that were observed or not observed, are shown in Figure 4 and Figure S15-S34. The average properties when not considering these groups/clusters is shown in Figure S34.

After considering all these types of clustering methods in different chemical spaces, one of the strongest trends between molecules observed and not observed was that molecules in ChemSpace Cluster 1 were not observed often. This can be found when using relative difference (Figure S11), where this cluster has a value of −166% and is a fairly large cluster with 94 members. From this group, only 8 members were observed and 86 (10.8 times more) were not observed. Average properties of these compounds show, relative to all spiked in compounds, a lower logP (partition coefficient) (0.86 vs 2.75, p-value: 8.6E-23) (Figure 4). Additionally, average mass of this cluster is significantly lower than masses of all spiked in compounds (159.8 vs. 260.8, p-value: 1.9E-26). This does give a good example where inherent ionizability is difficult to decouple from other experimental factors, such as matrix effects and instrument response, as compounds with low mass and high hydrophilicity typically have lower ESI activity relative to larger and more hydrophobic molecules.[40–43] Ionizability-specific trends have been closely investigated previously by Liigand et al.[44] This study reported the ionization of over 400 compounds across varying concentration, solvent, and instrument types and provides a machine learning-based tool to predict the ionization propensity of a given compound and the concentration of compounds in a solution based on the observed experimental data. Chemical descriptors they found to be most influential on their model were number of hydrogen atoms, number of nitrogen atoms, pH, viscosity, and presence of ammonium ions. These are the types of properties that can be utilized in *chespa* to analyze ionizability if desired by the user.

Overall, these different chemical spaces appear to enable the analysis of observability trends to varying degrees. ClassyFire Superclass appears to contribute the least, but additional ontological information could assist here. The analysis of potential trends due to chemical properties (as discussed above) is interesting and could be much more fruitful with additional properties available. This is not done here due to a more comprehensive dataset

needed to really assess varying trends. Additional discussion about the trends seen here are available in the SI (SI RD2).

CONCLUSIONS

Observability of molecules by MS is a challenging property to predict based on molecular structure, sample matrix, and analysis conditions, yet it has the potential to aid in lowering false discovery rates associated with small molecule identification and, in the case of targeted studies, reducing experimentation cost and time. Understanding the factors that contribute to observability could improve current compound identification techniques, especially in complex samples that can have 10–100's of thousands of unique molecules. With an ability to predict what compounds can be observed given some set of experimental conditions, libraries of candidate compounds could be pruned to remove molecules if they have a low chance of being observable. The hope is to decrease the false positive rate during small molecule identification steps (and improve true positive rates) to increase the value of the data used to assess hypotheses regarding the analyzed samples. Additionally, considering targeted methods, experimental set up (such as chosen ionization mode) could be selected for a given set of molecules of interest, reducing the cost of experimentation and focusing efforts where they are most beneficial. Due to the many variables that control observability, many questions still remain, including which compounds will be observable given a specific matrix, what properties of their molecular structure contribute to this, and, if they are observed in an ionized form, which adducts do they form? To date, software tools are nascent and do not provide automated functionality required to begin answering these questions.

To evaluate this important property, we developed a Python module, *chespa*, that can be used to streamline the chemical space analysis and assess and compare observability trends. To demonstrate an example use of *chespa*, we used data generated during our participation in the ENTACT challenge.[27, 45] While these initial results shown here are only relevant to our data in the context of the ENTACT challenge, we show how *chespa* could be used in subsequent studies to begin finding the critical features of molecules as they relate to observability or potentially other relevant properties. We imagine that with a sufficiently large dataset, with a wide variability in molecules and sample matrix types, *chespa* can be used to eventually build a predictive model for observability. Furthermore, *chespa* was built in a modular manner so that additional properties or cluster types can be added.

For example, a future application of this module can be to establish set clusters using a library with billions of compounds to define them. Along this line, using many predicted properties (versus the 10 used here) would be much more powerful, as it would provide more information about the trends seen in observability and make this module even more powerful. Pairing this with a larger set of experimentally-determined observability data would allow for binning of compounds into clusters based on predicted properties and used to predict whether they will be observable or not, and even in different conditions such as ionization mode. This could lead to lower false discovery rates and more efficient experimentation. Once these clusters are formed, using visualization software like

WebMolCS[17] could further assist in analysis, as this tool helps visualize chemical space in an interactive way.

Additionally, as mentioned, this module could be used to help assess other experimentally-measurable or predicted properties, such as toxicity, ligand affinity, and blood-brain barrier permeability. One example would be expanding the information used to predict ligand-based affinity for specific receptors, which are often based on a small set of input parameters.[46]

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

ACKNOWLEDGMENTS

This research was supported by the National Institutes of Health, National Institute of Environmental Health Sciences grant U2CES030170. Additional support was provided by the Laboratory Directed Research and Development program at Pacific Northwest National Laboratory (PNNL) and is a contribution of the Synthetic Biology (SynBio) Agile Project. This work was performed in the W. R. Wiley Environmental Molecular Sciences Laboratory (EMSL), a DOE national scientific user facility at the PNNL. PNNL is operated for DOE by Battelle Memorial Institute under contract DE-AC05-76RL01830.

REFERENCES

1. Dobson CM, Chemical space and biology. Nature 2004, 432, 824–828. [PubMed: 15602547]
2. Colby SM; Thomas DG; Nunez JR; Baxter DJ; Glaesemann KR; Brown JM; Pirrung MA; Govind N; Teeguarden JG; Metz TO; Renslow RS, Isicle: A quantum chemistry pipeline for establishing in silico collision cross section libraries. Anal. Chem. 2019, 91, 4346–4356. [PubMed: 30741529]
3. Moriwaki H; Tian Y-S; Kawashita N; Takagi T, Mordred: A molecular descriptor calculator. J. Cheminform. 2018, 10, 4. [PubMed: 29411163]
4. Yesiltepe Y; Nunez JR; Colby SM; Thomas DG; Borkum MI; Reardon PN; Washton NM; Metz TO; Teeguarden JG; Govind N; Renslow RS, An automated framework for nmr chemical shift calculations of small organic molecules. J. Cheminform. 2018, 10, 52. [PubMed: 30367288]
5. Bohacek RS; McMartin C; Guida WC, The art and practice of structure-based drug design: A molecular modeling perspective. 1996, 16, 3–50.
6. Polishchuk PG; Madzhidov TI; Varnek A, Estimation of the size of drug-like chemical space based on gdb-17 data. J. Comput.-Aided Mol. Des. 2013, 27, 675–679. [PubMed: 23963658]
7. Hamdalla MA; Mandoiu II; Hill DW; Rajasekaran S; Grant DF, Biosm: Metabolomics tool for identifying endogenous mammalian biochemical structures in chemical structure space. J. Chem. Inf. Model. 2013, 53, 601–612. [PubMed: 23330685]
8. Hartenfeller M; Schneider G, De novo drug design. In Chemoinformatics and computational chemical biology, Bajorath J, Ed. Humana Press: Totowa, NJ, 2011; pp 299–323.
9. Fourches D; Kuchibhotla S; Zin PP; Kuenemann MA, Cheminformatics modeling of closantel analogues for treating river blindness. ChemRxiv 2020.
10. van Deursen R; Blum LC; Reymond J-L, Visualisation of the chemical space of fragments, lead-like and drug-like molecules in pubchem. J. Comput.-Aided Mol. Des. 2011, 25, 649–662. [PubMed: 21618008]
11. Lim J; Ryu S; Kim JW; Kim WY, Molecular generative model based on conditional variational autoencoder for de novo molecular design. J Cheminform 2018, 10, 31. [PubMed: 29995272]
12. Li X; Xu Y; Yao H; Lin K, Chemical space exploration based on recurrent neural networks: Applications in discovering kinase inhibitors. Journal of Cheminformatics 2020, 12. [PubMed: 33431043]
13. Takeda S; Kaneko H; Funatsu K, Chemical-space-based de novo design method to generate drug-like molecules. J. Chem. Inf. Model. 2016, 56, 1885–1893. [PubMed: 27632418]

14. Allen CHG; Koutsoukas A; Cortes-Ciriano I; Murrell DS; Malliavin TE; Glen RC; Bender A, Improving the prediction of organism-level toxicity through integration of chemical, protein target and cytotoxicity qhts data. Toxicol Res (Camb) 2016, 5, 883–894. [PubMed: 30090397]
15. Boyd SM; de Kloe GE, Fragment library design: Efficiently hunting drugs in chemical space. Drug Discovery Today: Technologies 2010, 7, e173–e180.
16. Rasche F; Scheubert K; Hufsky F; Zichner T; Kai M; Svatoš A; Böcker S, Identifying the unknowns by aligning fragmentation trees. Anal. Chem. 2012, 84, 3417–3426. [PubMed: 22390817]
17. Awale M; Probst D; Reymond J-L, Webmolcs: A web-based interface for visualizing molecules in three-dimensional chemical spaces. Journal of Chemical Information and Modeling 2017, 57, 643–649. [PubMed: 28316236]
18. Colby SM; Nuñez JR; Hodas NO; Corley CD; Renslow RR, Deep learning to generate in silico chemical property libraries and candidate molecules for small molecule identification in complex samples. Anal Chem 2020, 92, 1720–1729. [PubMed: 31661259]
19. Maziarka Ł; Pocha A; Kaczmarczyk J; Rataj K; Danel T; Warchoł M, Mol-cyclegan: A generative model for molecular optimization. Journal of Cheminformatics 2020, 12. [PubMed: 33431043]
20. Mariya Popova OI, Alexander Tropsha, Deep reinforcement learning for de novo drug design. Sci. Adv. 2018, 4.
21. Virshup AM; Contreras-Garcia J; Wipf P; Yang W; Beratan DN, Stochastic voyages into uncharted chemical space produce a representative library of all possible drug-like compounds. J Am Chem Soc 2013, 135, 7296–303. [PubMed: 23548177]
22. Real space navigator version 4.0; biosolveit gmbh, sankt; biosolveit.De/realspacenavigator.
23. O'Boyle NM; Banck M; James CA; Morley C; Vandermeersch T; Hutchison GR, Open babel: An open chemical toolbox. J. Cheminform. 2011, 3, 33. [PubMed: 21982300]
24. Yesiltepe Y; Renslow RS, Spectre. arXiv 2020.
25. Djoumbou Feunang Y; Eisner R; Knox C; Chepelev L; Hastings J; Owen G; Fahy E; Steinbeck C; Subramanian S; Bolton E; Greiner R; Wishart DS, Classyfire: Automated chemical classification with a comprehensive, computable taxonomy. J. Cheminform. 2016, 8, 61. [PubMed: 27867422]
26. Ulrich EM; Sobus JR; Grulke CM; Richard AM; Newton SR; Strynar MJ; Mansouri K; Williams AJ, Epa's non-targeted analysis collaborative trial (entact): Genesis, design, and initial findings. Analytical and Bioanalytical Chemistry 2019, 411, 853–866. [PubMed: 30519961]
27. Nuñez JR; Colby SM; Thomas DG; Tfaily MM; Tolic N; Ulrich EM; Sobus JR; Metz TO; Teeguarden JG; Renslow RS, Evaluation of in silico multifeature libraries for providing evidence for the presence of small molecules in synthetic blinded samples. Journal of Chemical Information and Modeling 2019, 59, 4052–4060. [PubMed: 31430141]
28. Jupyter P; Bussonnier M; Forde J; Free-man J; Granger B; Head T; Holdgraf C; Kelley K; Nalvarte G; Osheroff A; Pacer M; Panda Y; Perez F; Ragan-Kelley B; Willing C. In Binder 2.0 - reproducible, interactive, sharable environments for science at scale, Proceedings of the 17th Python in Science Conference, 2018; pp 113–120.
29. Walt S. v. d.; Colbert SC; Varoquaux G, The numpy array: A structure for efficient numerical computation. Computing in Science & Engineering 2011, 13, 22–30.
30. Pedregosa F; Varoquaux G; Gramfort A; Michel V; Thirion B; Grisel O; Blondel M; Prettenhofer P; Weiss R; Dubourg V; Vanderplas J; Passos A; Cournapeau D; Brucher M; Perrot M; Duchesnay E, Scikit-learn: Machine learning in python. J. Mach. Learn. Res. 2011, 12, 2825–2830.
31. Sprock C. Bmdcluster (v0.3.1).
32. Jeff Reback; Wes McKinney; Joris Van den Bossche; jbrockmendel; Tom Augspurger; Phillip Cloud; gfyoung; Sinhrks; Adam Klein; Jeff Tratner; Chang She; Matthew Roeschke; Terji Petersen; William Ayd; Andy Hayden; Simon Hawkins; Jeremy Schendel; Marc Garcia; Vytautas Jancauskas; Pietro Battiston; Skipper Seabold; chris-b1; h-vetinari; Stephan Hoyer; Wouter Overmeire; Mortada Mehyar; behzad nouri; Thomas Kluyver; Christopher Whelan; Chen KW. Pandas-dev/pandas: Pandas 0.25.1.
33. RDKit: Open-source cheminformatics. rdkit.org.
34. Michael Waskom; Olga Botvinnik; Paul Hobson; John B. Cole; Yaroslav Halchenko; Stephan Hoyer; Alistair Miles; Tom Augspurger; Tal Yarkoni; Tobias Megies; Luis Pedro Coelho; Daniel

Wehner; cynddl; Erik Ziegler; diego0020; Yury V Zaytsev; Travis Hoppe; Skipper Seabold; Phillip Cloud; Miikka Koskinen; Kyle Meyer; Adel Qalieh; Allan D. Seaborn: V0.9.0 (july 2018).

35. Hunter JD, Matplotlib: A 2d graphics environment. Computing in Science & Engineering 2007, 9, 90–95.
36. May JC; Goodwin CR; Lareau NM; Leaptrot KL; Morris CB; Kurulugama RT; Mordehai A; Klein C; Barry W; Darland E; Overney G; Imatani K; Stafford GC; Fjeldsted JC; McLean JA, Conformational ordering of biomolecules in the gas phase: Nitrogen collision cross sections measured on a prototype high resolution drift tube ion mobility-mass spectrometer. Anal. Chem. 2014, 86, 2107–2116. [PubMed: 24446877]
37. Ibrahim YM; Baker ES; Danielson WF 3rd; Norheim RV; Prior DC; Anderson GA; Belov ME; Smith RD, Development of a new ion mobility (quadrupole) time-of-flight mass spectrometer. Int. J. Mass spectrom. 2015, 377, 655–662. [PubMed: 26185483]
38. Arthur D; Vassilvitskii S, K-means++: The advantages of careful seeding. In Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, Society for Industrial and Applied Mathematics: New Orleans, Louisiana, 2007; pp 1027–1035.
39. Rousseeuw PJ, Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics 1987, 20, 53–65.
40. Null AP; Nepomuceno AI; Muddiman DC, Implications of hydrophobicity and free energy of solvation for characterization of nucleic acids by electrospray ionization mass spectrometry. Anal. Chem. 2003, 75, 1331–1339. [PubMed: 12659193]
41. Mirzaei H; Regnier F, Enhancing electrospray ionization efficiency of peptides by derivatization. Anal. Chem. 2006, 78, 4175–83. [PubMed: 16771548]
42. Cech NB; Enke CG, Relating electrospray ionization response to nonpolar character of small peptides. Anal. Chem. 2000, 72, 2717–2723. [PubMed: 10905298]
43. Fenn JB, Ion formation from charged droplets: Roles of geometry, energy, and time. J. Am. Soc. Mass. Spectrom. 1993, 4, 524–535. [PubMed: 24227639]
44. Liigand J; Wang T; Kellogg J; Smedsgaard J; Cech N; Kruve A, Quantification for non-targeted lc/ms screening without standard substances. Scientific Reports 2020, 10, 5808. [PubMed: 32242073]
45. Sobus JR; Wambaugh JF; Isaacs KK; Williams AJ; McEachran AD; Richard AM; Grulke CM; Ulrich EM; Rager JE; Strynar MJ; Newton SR, Integrating tools for non-targeted analysis research and chemical safety evaluations at the us epa. J. Expo. Sci. Environ. Epidemiol. 2017, 28, 411–426. [PubMed: 29288256]
46. Schultz K; Colby SM; Yesiltepe Y; Nunez J; Mcgrady MY; Renslow RS, Application and assessment of deep learning to generate potential nmda receptor antagonists. Chemical Science (in review) 2020.

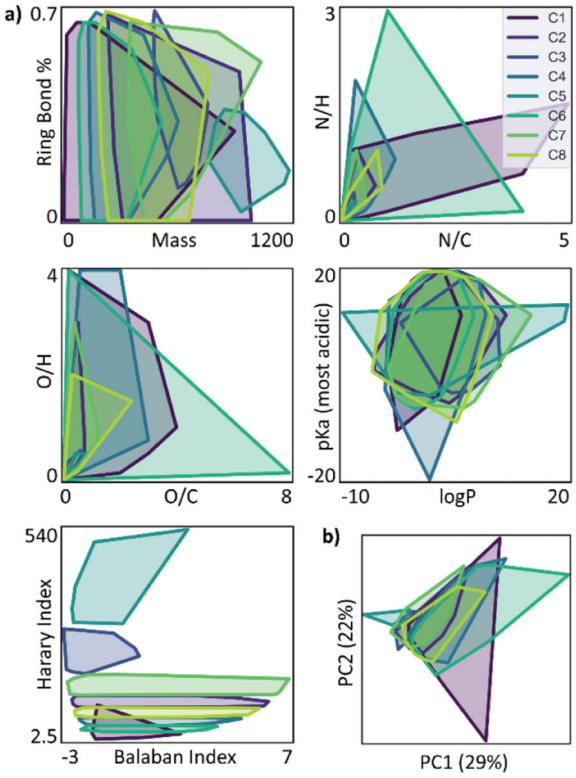


Figure 1. Chemical space covered by each KMeans cluster.

(a) Distribution of predicted properties. (b) PC1 and 2 from the principal component analysis performed on the properties plotted in (a).

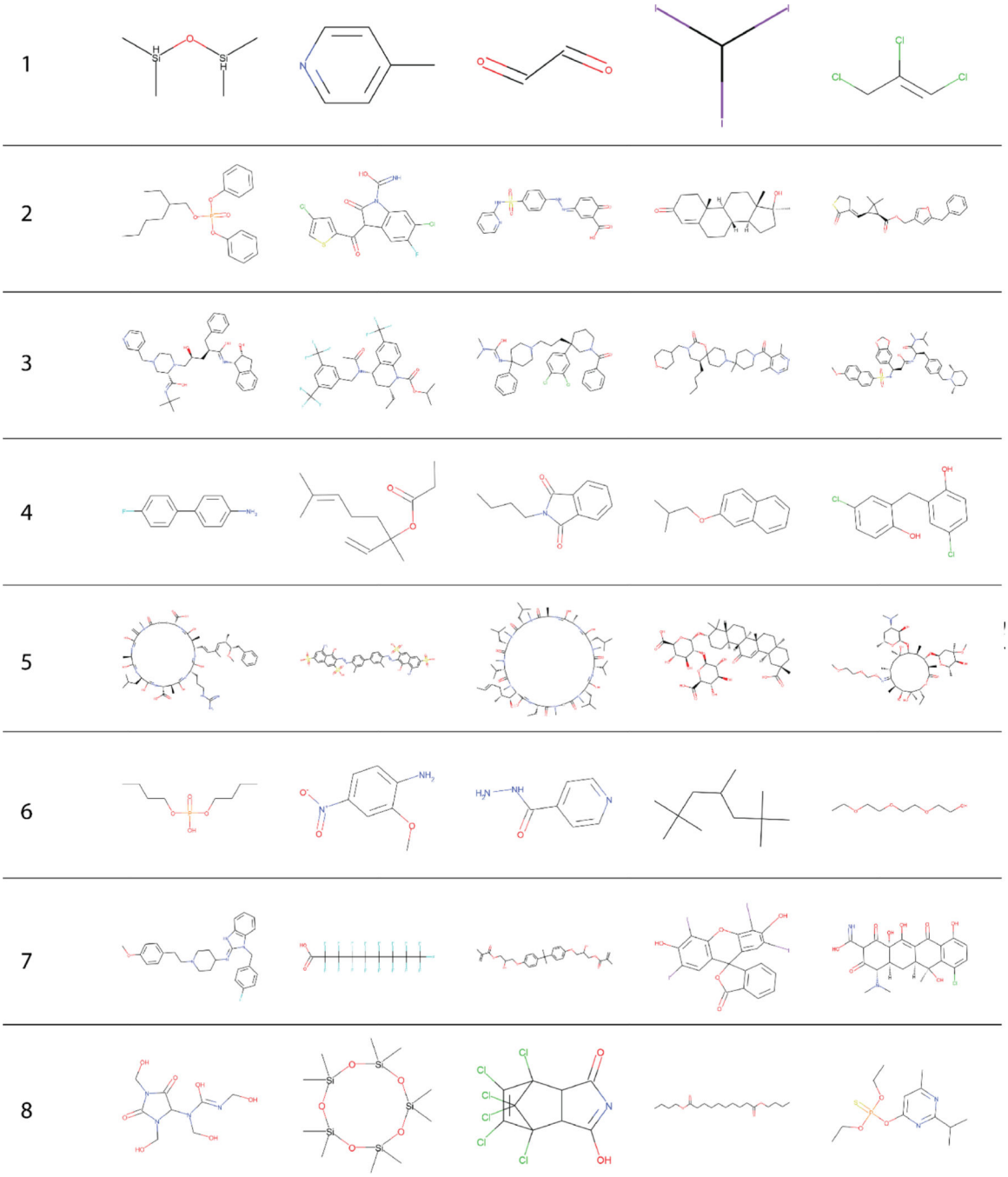


Figure 2.
Five randomly chosen compounds in each of the 8 Chemical Space clusters.

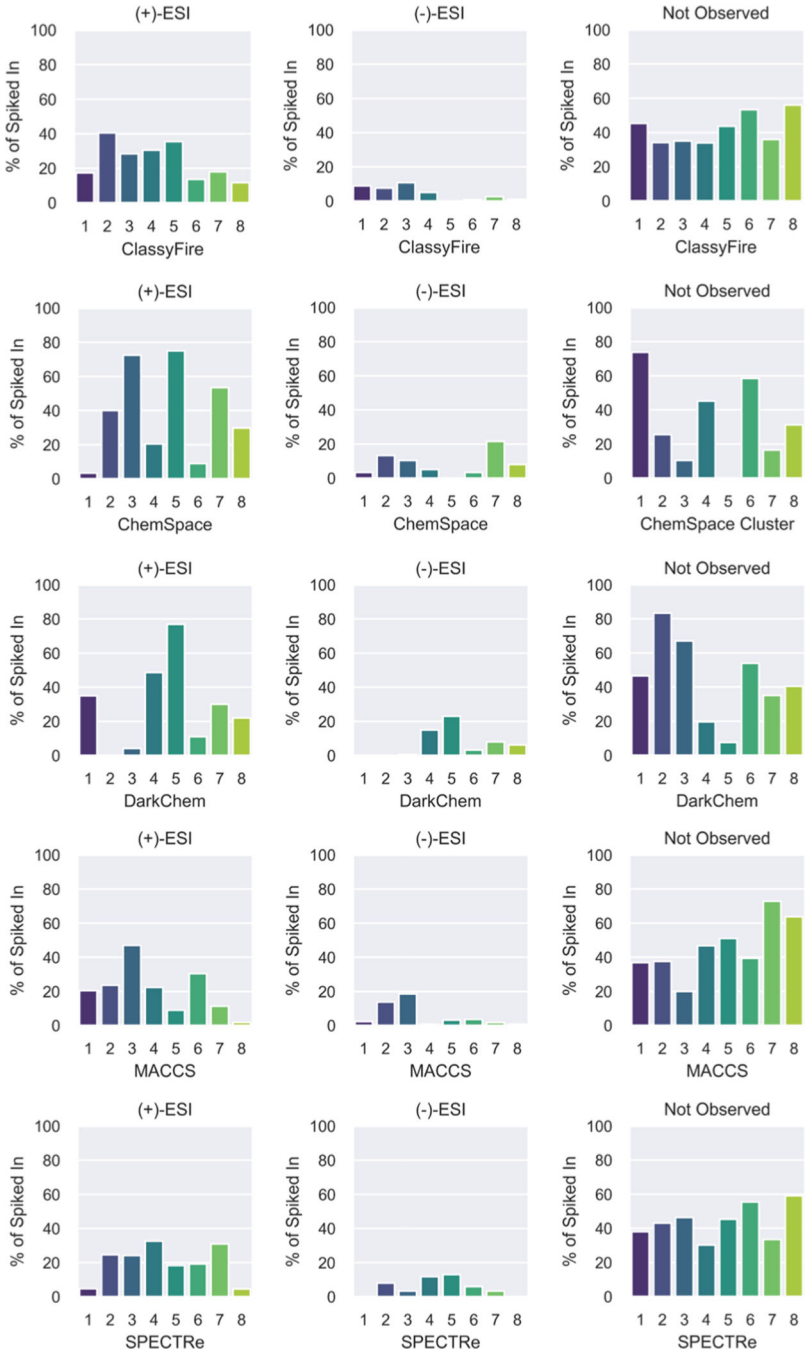


Figure 3. Distribution of spiked in compounds observed using positive vs. negative mode ESI, and the distribution of those that were not observed in any sample.

Compounds were split into groups by ClassyFire superclass groups, ChemSpace clusters, DarkChem clusters, and substructure (MACCS and SPECTRe) clusters.

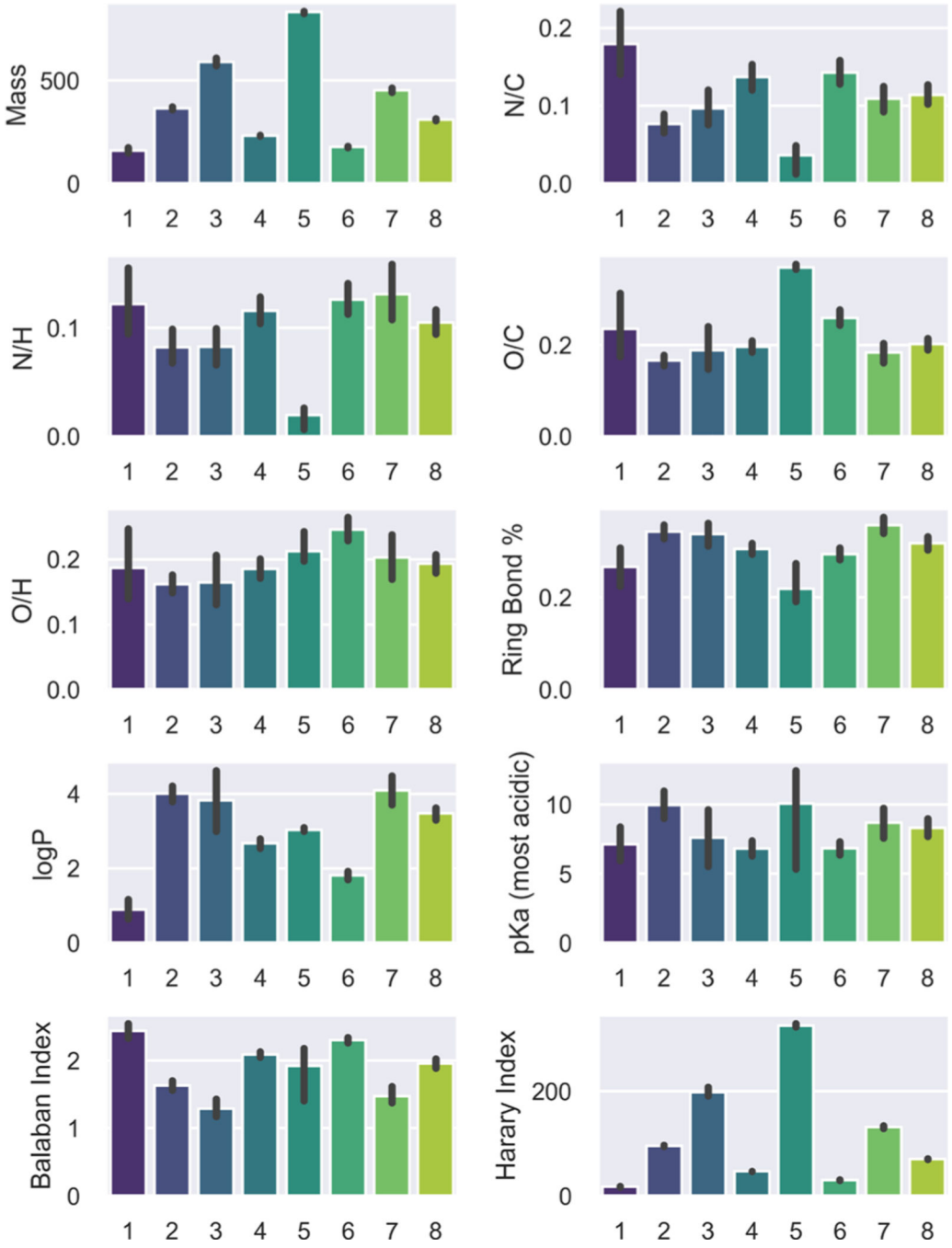


Figure 4. Average properties of compounds in Chemical Space clusters, considering only spiked in compounds.

Error bars represent the standard deviation.