

1 cohorts: A Python package for clinical 'omics 2 data management

3 Nicholas P. Giangreco^{1,2}, Barry Fine³ and Nicholas P. Tatonetti*^{1,2}

4 Department of ¹Systems Biology and ²Biomedical Informatics, ³Division of Cardiology, Columbia University,
5 622 W. 168th Street, New York, NY, 10032.

6 Abstract

7 **Summary:** Precision medicine uses patient clinical and molecular characteristics to personalize diagnosis
8 and treatment. This emerging discipline integrates multi-modal data into large-scale studies of human
9 disease to make accurate individual-level predictions. The success of these studies will depend on the
10 generalizability of the results, the ability of other researchers and clinicians to replicate studies, and the
11 understandability of the methods used. Tools for data management and standardization are needed to
12 promote flexible, transparent, and reproducible analyses. Here we present *cohorts*, a python package
13 facilitating clinical and biomarker data management to enhance standardization and reproducibility of
14 clinical findings.

15 **Availability:** The python package *cohorts* is available at <http://www.github.com/ngiangre/cohorts>.

16 **CONTACT:** NPT2105@CUMC.COLUMBIA.EDU

17

18 1. Introduction

19 Precision medicine improves patient diagnoses and treatments by integrating clinical and biomarker data, which
20 requires effective data management for reproducibility of clinical research findings (Vargas and Harris, 2016; Leopold
21 and Loscalzo, 2018; Niven *et al.*, 2018). The studies often span institutions, departments, and research teams, where
22 patient data is collected and processed in a specialized way. Clinical studies including patient data from multiple
23 sources are integrated and managed by the primary research team, and so correct attribution of the clinical and
24 biomarker characteristics is critical. Feasible and accessible software can facilitate data storage and management,

25 which promotes flexible, transparent, and reproducible analyses. Moreover, software that facilitates integration of
26 multiple patient cohorts can promote the use of advanced statistical and machine learning analyses.

27 Precision medicine calls for rigor and reproducibility in data generation and data analysis (Mueller *et al.*, 2018;
28 Orton and Doucette, 2013). For large consortium based studies, computational platforms allow for sharing data and
29 merging of multisite datasets (Lam *et al.*, 2016). While large-scale studies may have an extensive computing platform
30 with custom software (Wolstencroft *et al.*, 2015; Price *et al.*, 2019), most clinical studies are smaller and would benefit
31 from standardized procedures and interoperable data management and analysis tools. There are many software
32 packages and platforms that perform precision medicine data analysis, such as pyGeno (Daouda *et al.*, 2016) and in
33 particular packages from the package manager Bioconductor in the R programming language (Huber *et al.*, 2015).
34 However, many of these applications can handle only one experimental type or are particular to a specific end-to-end
35 data analysis. There is a need for flexible and open source software, where patient clinical and biomarker data can
36 originate from different cohorts, platforms, and data types.

37 Here, we present the python package *cohorts*, which provides storage and management of patient clinical and
38 biomarker data. Moreover, *cohorts* facilitates integration of multiple patient cohort data through common attributes
39 shared by each cohort. *Cohorts* is light-weight and portable software promoting standardized and reproducible clinical
40 research analyses. We provide this open source python package on Github at <http://www.github.com/ngiangre/cohorts>.

41 2. Results

42 2.1 Package overview

43 The main objective of *cohorts* is for the storage and management of patient clinical and biomarker datasets (Fig. 1).
44 The management of patient data is through the python class **Cohort** (classes are denoted in bold). **Cohort** takes as
45 input file locations of 1) a replicate or sample dataframe, where rows are biomarkers and columns are replicates or
46 samples (names of rows and columns are marker ids and replicate or sample names, respectively), and 2) a sample
47 groups dataframe, where rows are the clinical outcomes or groups (e.g. reference and treatment; can be any data type)
48 and columns are the samples (names of rows and columns are the group names and the sample names, respectively).
49 In addition to the name of the cohort and additional parameters (see package for details), **Cohort** derives and stores
50 attributes of the dataset in an object-oriented fashion, containing both common attributes to other declared classes of
51 **Cohort** as well as specific attributes for the patient cohort to be used in downstream analysis. The data management
52 by the patient cohort object allows for standardized analysis with minimal initial data upload and wrangling. Moreover,
53 the python object gives consistent and logical data attributes that are easily retrievable for downstream analyses.

54 Integration of multiple datasets is through the **IntegratedCohort** python class. By providing a dictionary of **Cohort**
 55 objects, **IntegratedCohort** parses, integrates, and containerizes shared information from each **Cohort** object. This
 56 class declaration is created by using the common data attributes of the **Cohort** class (at least one **Cohort** object is
 57 needed to produce an **IntegratedCohort** object). Additionally, specific attributes are preserved for the individual patient
 58 cohorts and are available for the newly created **IntegratedCohorts** object. Specific attributes for this object can
 59 facilitate, for example, managing data from statistical and machine learning tasks using shared biomarkers or clinical
 60 groups. Again, no initial data upload and minimal wrangling is needed for this integration task, saving the user time to
 61 focus on standardized and reproducible downstream biomarker analysis.

62 2.2 Implementation

63 Implementation of the *cohorts* package is described and shown in the supplementary material in the online version of
 64 the manuscript. Briefly, data for an individual patient cohort is managed through the **Cohorts** class. The patient data
 65 is retrieved through accessing object attributes, and new data can be stored within the data attribute. The
 66 **IntegratedCohorts** class integrates patient data stored in *Cohorts* objects using the “integrate_cohorts” method,
 67 which can access, parse, integrate, and store all the patient data within the **IntegratedCohort** class. Example
 68 procedures streamlined using the package include descriptive statistics of molecular marker data and outcome
 69 prediction using integrated patient data that share common attributes. An example can be found in a jupyter notebook
 70 at https://github.com/ngiangre/cohorts/blob/master/Bioinformatics_Note_Implementation.ipynb.

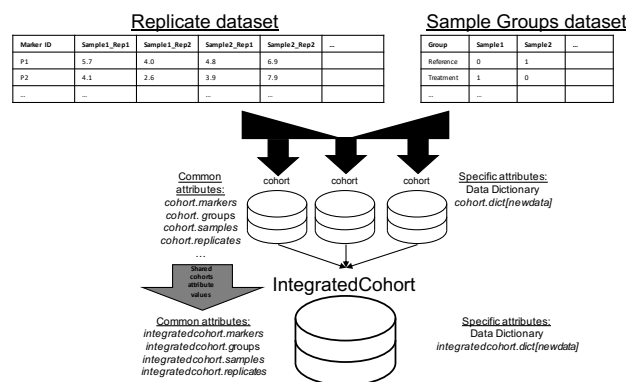


Figure 1: Schematic of data input, storage, management, and integration workflow for the *cohorts* package.

71 3. Conclusion

72 Studies in precision medicine need standardized and reproducible analyses. The *cohorts* package supports data
73 management, procedural standardization, and data integration for analyses that will improve the flexibility,
74 transparency and reproducibility of clinical investigations. We provide jupyter notebooks giving an overview and
75 usage of the package in the *cohorts* GitHub repository at <http://www.github.com/ngiangre/cohorts>.

76 **Acknowledgements**

77 Thank you to the Columbia University Irving Medical Center Proteomics Core for generation and pre-processing of patient proteomic samples.
78 Most importantly, thank you to the patients for consenting to partaking in clinical research.

79 **Funding**

80 This work is supported by the National Institute of General Medical Sciences [R01GM107145] and the Precision Medicine Award at the Irving
81 Institute for Clinical and Translational Research.

82 *Conflict of Interest:* none declared.

84 **References**

85 Daouda, T. *et al.* (2016) pyGeno: A Python package for precision medicine and proteogenomics. *F1000Research*, **5**,
86 381.

87 Huber, W. *et al.* (2015) Orchestrating high-throughput genomic analysis with Bioconductor. *Nat. Methods*, **12**, 115–
88 121.

89 Lam, R.W. *et al.* (2016) Discovering biomarkers for antidepressant response: Protocol from the Canadian biomarker
90 integration network in depression (CAN-BIND) and clinical characteristics of the first patient cohort. *BMC*
91 *Psychiatry*, **16**, 1–13.

92 Leopold, J.A. and Loscalzo, J. (2018) Emerging Role of Precision Medicine in Cardiovascular Disease. *Circ. Res.*, **122**,
93 1302–1315.

94 Mueller, C. *et al.* (2018) Protein biomarkers for subtyping breast cancer and implications for future research. *Expert*
95 *Rev. Proteomics*, **15**, 131–152.

96 Niven, D.J. *et al.* (2018) Reproducibility of clinical research in critical care: A scoping review. *BMC Med.*, **16**, 1–12.

97 Orton, D. and Doucette, A. (2013) Proteomic Workflows for Biomarker Identification Using Mass Spectrometry —
98 Technical and Statistical Considerations during Initial Discovery. *Proteomes*, **1**, 109–127.

99 Price, N. *et al.* (2019) A Scalable Data Science Platform for Healthcare and Precision Medicine Research (Preprint). *J.*
100 *Med. Internet Res.*

101 Vargas, A.J. and Harris, C.C. (2016) Biomarker development in the precision medicine era: Lung cancer as a case

102 study. *Nat. Rev. Cancer*, **16**, 525–537.

103 Wolstencroft, K. *et al.* (2015) SEEK: A systems biology data and model management platform. *BMC Syst. Biol.*, **9**, 1–
104 12.

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128 Supplementary Material

129 Implementation

130 Implementation of the *cohorts* package is described and shown below. The source code can be found at
131 https://github.com/ngiangre/cohorts/blob/master/Bioinformatics_Note_Implementation.ipynb.

132 Patient cohort data

133 The **Cohorts** class manages data from a single patient cohort. The package is accompanied by a synthetically
134 generated patient cohort dataset at https://github.com/ngiangre/cohorts/tree/master/sample_data. The data, which in
135 this example will include protein biomarker expression, follows the necessary format outlined in section 2.1. After
136 importing the *cohorts* package, the patient cohort data is uploaded by specifying directory and file locations as values
137 and the cohort name as the key within a dictionary. The dictionary is then given as a parameter to the **Cohort** class
138 constructor to instantiate a patient cohort data object. Within the **Cohort** class, the patient cohort data is automatically
139 parsed and stored as attributes in the newly created object. The patient data can then be retrieved through accessing
140 object attributes, and new data can be stored within the data attribute. For example, the distribution of protein
141 expression across patients can be accessed and displayed (Fig. 2).

142

143

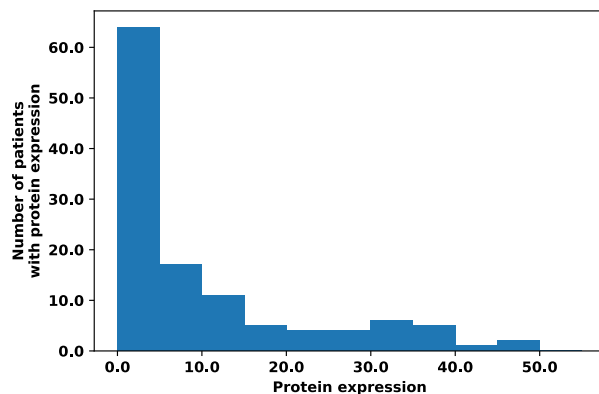


Figure 2. Protein expression distribution across patient cohort samples using the *cohort* package. See Implementation jupyter notebook for source code.

144 Integration of multiple patient cohort data

145 The **IntegratedCohorts** class integrates patient data stored in *Cohorts* objects. As shown in the notebook, a
146 dictionary of three different patient cohort datasets is input to create an instance of the **IntegratedCohorts** class.
147 After calling the 'integrate_cohorts' method the original patient cohort data is accessed, parsed, integrated, and
148 stored within attributes of the **IntegratedCohort** class. In addition to data integration, the individual patient cohort
149 data is stored separately upon instantiation. The integration of related patient biomarker and clinical data provides the
150 ability to generalize findings from an individual to a broader patient population. An example is the task of disease
151 prediction. The characterization of disease by biomarkers or clinical markers may only be representative in a patient
152 population, but providing similar results within a larger patient population provides stronger evidence for association
153 of patient markers to a disease. Figure 3 shows a receiver operator characteristic curve from predicting disease from
154 protein marker expression levels of patient samples in the three patient cohort datasets.
155

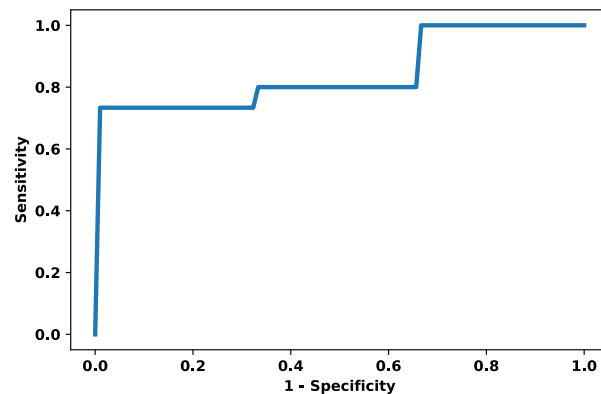


Figure 3. Receiver-operator characteristic curve for disease prediction using protein marker expression as features in a Logistic Regression classifier. The average sensitivity is shown from 5-fold cross validation.