SOFTWARE TOOL ARTICLE

The Neuroimaging Data Model Linear Regression Tool (nidm_linreg): PyNIDM Project [version 1; peer review: 1 approved with reservations, 1 not approved]

Ashmita Kumar ¹, Albert Crowley ², Nazek Queder ³, JB Poline⁴,
Satrajit S. Ghosh ^{5,6}, David Kennedy ⁷, Jeffrey S. Grethe ⁸, Karl G. Helmer ⁹,
David B. Keator³

¹Troy High School, Fullerton, California, USA

²TCG, Inc., Washington, DC, USA

³Psychiatry and Human Behavior, University of California, Irvine, Irvine, California, USA

⁴Department of Neurology and Neurosurgery, McConnell Brain Imaging Centre, McGill University, Faculty of Medicine and Health Sciences, Montreal, Canada

⁵McGovern Institute for Brain Research, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA

⁶Department of Otolaryngology, Harvard Medical School, Boston, Massachusetts, USA

⁷Department of Psychiatry, Eunice Kennedy Shriver Center, University of Massachusetts Chan Medical School, Worcester, Massachusetts, USA

⁸Department of Neuroscience, Harvard Medical School, San Diego, California, USA

⁹Department of Radiology, Massachusetts General Hospital, Harvard Medical School, Boston, Massachusetts, USA

V1 First published: 24 Feb 2022, 11:228
<https://doi.org/10.12688/f1000research.108008.1>
Latest published: 29 Jul 2022, 11:228
<https://doi.org/10.12688/f1000research.108008.2>

Abstract

The Neuroimaging Data Model (NIDM) is a series of specifications for describing all aspects of the neuroimaging data lifecycle from raw data to analyses and provenance. NIDM uses community-driven terminologies along with unambiguous data dictionaries within a Resource Description Framework (RDF) document to describe data and metadata for integration and query. Data from different studies, using locally defined variable names, can be retrieved by linking them to higher-order concepts from established ontologies and terminologies. Through these capabilities, NIDM documents are expected to improve reproducibility and facilitate data discovery and reuse. PyNIDM is a Python toolbox supporting the creation, manipulation, and querying of NIDM documents. Using the query tools available in PyNIDM, users are able to interrogate datasets to find studies that have collected variables measuring similar phenotypic properties. This, in turn, facilitates the transformation and combination of data across multiple studies.

The focus of this manuscript is the linear regression tool which is a part of the PyNIDM toolbox and works directly on NIDM documents. It provides a high-level statistical analysis that aids researchers in

Open Peer Review

Approval Status

	1	2
version 2 (revision) 29 Jul 2022	view

version 1 24 Feb 2022	view	view

1. **Adam G. Thomas** , National Institute of Mental Health, Bethesda, USA

Eric Earl , National Institute of Mental Health, Bethesda, USA

2. **Karsten Specht** , University of Bergen, Bergen, Norway

Any reports and responses or comments on the

gaining more insight into the data that they are considering combining across studies. This saves researchers valuable time and effort while showing potential relationships between variables. The linear regression tool operates through a command-line interface integrated with the other tools (pynidm linear-regression) and provides the user with the opportunity to specify variables of interest using the rich query techniques available for NIDM documents and then conduct a linear regression with optional contrast and regularization.

article can be found at the end of the article.

Keywords

Linear Regression, Neuroimaging, PyNIDM, Neuroimaging Data Model, Machine Learning

This article is included in the **INCF** gateway.

This article is included in the **Artificial Intelligence and Machine Learning** gateway.

Corresponding author: Ashmita Kumar (ashmita.kumar@gmail.com)

Author roles: **Kumar A:** Conceptualization, Formal Analysis, Investigation, Methodology, Project Administration, Software, Validation, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; **Crowley A:** Formal Analysis, Resources, Software, Validation, Visualization; **Queder N:** Formal Analysis, Writing – Review & Editing; **Poline J:** Formal Analysis, Resources, Validation, Writing – Review & Editing; **Ghosh SS:** Formal Analysis, Resources, Validation, Writing – Review & Editing; **Kennedy D:** Formal Analysis, Writing – Review & Editing; **Grethe JS:** Formal Analysis, Writing – Review & Editing; **Helmer KG:** Formal Analysis, Writing – Review & Editing; **Keator DB:** Data Curation, Formal Analysis, Funding Acquisition, Project Administration, Resources, Software, Supervision, Validation, Writing – Original Draft Preparation, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: This work has been supported by the National Institute of Mental Health under grant RF1 MH120021 (PI: Keator), the International Neuroinformatics Coordinating Facility (INCF), and from the National Institute of Biomedical Imaging and Bioengineering P41 EB019936 (PI: Kennedy).

Copyright: © 2022 Kumar A *et al.* This is an open access article distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Kumar A, Crowley A, Queder N *et al.* **The Neuroimaging Data Model Linear Regression Tool (nidm_linreg): PyNIDM Project [version 1; peer review: 1 approved with reservations, 1 not approved]** F1000Research 2022, 11:228 <https://doi.org/10.12688/f1000research.108008.1>

First published: 24 Feb 2022, 11:228 <https://doi.org/10.12688/f1000research.108008.1>

Introduction

Background

The Neuroimaging Data Model (NIDM) (Keator *et al.* 2013; NIDM Working Group; Maumet *et al.* 2016) (Neuroimaging Data Model, RRID:SCR_013667) was started by an international team of volunteers to create specifications for describing all aspects of the neuroimaging data lifecycle. NIDM is built upon the PROV Standard (Moreau *et al.* 2008; “PROV-Overview”). It consists of three specifications: Experiment, Results, and Workflow. Using semantic web techniques (“Semantic Web - W3C”), these specifications were envisioned to capture information on all aspects of the neuroimaging data lifecycle, producing graphs linking each result’s artifact with the workflow that produced it and the data used in the computation. These graphs can be serialized into a variety of text-based formats (NIDM documents), and with the capabilities of the semantic web, can be used to link datasets together through annotations with terms from formal terminologies, complete data dictionaries of study variables, and linkage of study variables to broader concepts. These annotations provide a critical capability to aid in reproducibility and replication of studies, as well as data discovery in shared resources. The NIDM-Experiment model consists of a simple project-session-acquisition hierarchy which can be used to describe both the content and metadata about experimental studies and derived (e.g., regional brain volume, mass-univariate functional brain analysis) neuroimaging data. It has been used to describe many large publicly-available human neuroimaging datasets (e.g. ABIDE (Di Martino *et al.* 2014), ADHD200 (Milham *et al.* 2011), CoRR (Zuo *et al.* 2014) (Consortium for Reliability and Reproducibility, RRID:SCR_003774), OpenNeuro (“OpenNeuro”) (OpenNeuro, RRID:SCR_005031) datasets) along with providing unambiguous descriptions of the clinical, neuropsychological, and imaging data collected as part of those studies.

PyNIDM (*PyNIDM*) (PyNIDM, RRID:SCR_021022) v3.9.5 is a Python toolbox under active development that supports the creation, manipulation, and query of NIDM documents. It is open-source and hosted on GitHub, distributed under the Apache License, Version 2.0 (“Apache License, Version 2.0”). PyNIDM consists of tools to work with NIDM documents such as conversion from BIDS (Gorgolewski *et al.* 2016), graph visualization, serialization format conversion, merging and query. Querying of NIDM documents is supported using a command-line RESTful (Ravan *et al.* 2020) interface (i.e. pynidm query) which executes SPARQL (“SPARQL Query Language for RDF”) queries. Using the query functionality and the NIDM document semantics, users can quickly identify datasets that measured similar properties and may be combined for further investigation.

Beyond the existing tools that have been written to support NIDM documents, some high-level statistical analysis tools are needed to provide investigators with an opportunity to gain more insight into data they may be interested in combining for a complete scientific investigation. Combining datasets collected across different studies is not a trivial task. It requires both a complete, unambiguous description of the data and how it was collected, along with a varying number of transformations to align, where possible, disparate data. The process of transforming data is often quite time-consuming and therefore understanding whether the identified datasets, at a high level, might have some interesting relationships prior to committing to a full scientific study is prudent. Here we report on a tool that provides such capabilities; namely, a simple linear modeling tool supporting NIDM documents and integrated into the existing PyNIDM suite of tools.

Statement of need

While tools and libraries for statistics and machine learning algorithms are numerous, there are none that can be directly applied to NIDM documents. The linear regression algorithm presented here allows scientists studying the human brain to easily find relationships between variables across datasets while retaining the provenance present in NIDM documents. The algorithm has the ability to query for specific variables or across similar variables from different studies using concept annotations on the variables. It then provides the user with the ability to construct arbitrary linear models on those data, supporting interactions between variables, contrasts of learned parameter sets, and L1 and L2 regularization (Nagpal 2017). There is no comparable tool for this use case.

Methods

Implementation and use cases

The linear regression tool, *nidm_linreg*, uses the PyNIDM query functionality to aggregate data in NIDM documents serialized using the standard Terse Resource Description Framework (RDF) Triple Language (TURTLE) (“RDF 1.1 Turtle”), a common semantic-web serialization format that is both structured for ease of use with computers and relatively easy for humans to read. Researchers have the ability to construct custom models based on their scientific goals. The source code is available on Zenodo and full details can be found in the Software Availability statement (Keator *et al.* 2021).

Thus, *nidm_linreg* is a machine learning algorithm that can work on complex datasets described using the NIDM linked-data format, while being reasonably easy to use. Researchers have the ability to conduct a preliminary analysis to understand if it is worth the effort to pursue combining datasets and doing the transformations necessary to integrate those

datasets. One can quickly determine if there are high-level relationships in the datasets and look at the different weights to decide what variables may warrant further study.

The tool provides a simple command-line user interface (Figure 1) based on the “Click” Python library (“Welcome to Click — Click Documentation (8.0.X)”) which integrates the linear regression module with existing PyNIDM tools (e.g. pynidm linear-regression, pynidm query, pynidm convert, etc.).

To use the tool, the user runs the command `pynidm linear-regression` with a variety of required and optional parameters. The first parameter, “-nl”, is a comma-separated list of NIDM serialized TURTLE files, each representing a single dataset or a collection site within a multi-site research project or multiple datasets (Figure 2). A useful set of NIDM documents describing publicly-available neuroimaging data from the ABIDE, ADHD200, and CoRR studies along with datasets in the OpenNeuro database can be found on [GitHub \(D. Keator\)](#). The next parameter, “-model” provides the user with the ability to construct a linear model using notation found in popular statistics packages (e.g., R statistical software (Ripley 2001) (R Project for Statistical Computing, RRID:SCR_001905)). The syntax follows the scheme “dependent variable (DV) = independent variable 1 (IV1) + independent variable 2 (IV2) + ... + IVX”. To encode interactions between IV1 and IV2 in the above example, one can use the common “*” syntax: “DV = IV1 + IV2 + IV1*IV2”.

To determine what variables or data elements are available from a set of NIDM documents, the first step is to use “pynidm query” to do a data element search of the NIDM documents. From this search, the user can see what data elements are available in the selected NIDM documents and understand some details of those data elements (e.g., ranges, categories, data type, etc.). After performing the data elements query of the NIDM documents and selecting independent and dependent variables of interest, one proceeds with constructing the linear model with the pynidm linear-regression tool.

```
Options:
-nl, --nidm_file_list TEXT  A comma separated list of NIDM files with full
                             path [required]

-contrast, --ctr TEXT       This parameter will show differences in
                             relationship by group (e.g. -contrast
                             age*sex,group). It can be one variable,
                             interacting variables, or multiple

-model, --ml TEXT           This parameter will return the results of the
                             linear regression from all nidm files supplied
                             The way this looks in the command is python3
                             nidm_linreg.py -nl MTdemog_aseg_v2.ttl -model
                             "fs_003343 = age*sex + sex + age + group +
                             age*group + bmi" -contrast group -r L1
                             [required]

-o, --output_file TEXT      Optional output file (TXT) to store results of
                             the linear regression, contrast, and
                             regularization

-r, --regularization TEXT    This parameter will return the results of the
                             linear regression with L1 or L2 regularization
                             depending on the type specified, and the weight
                             with the maximum likelihood solution

--help                      Show this message and exit.
```

Figure 1. pynidm linear-regression parameters; demonstrating options for a researcher using the tool.

```
pynidm linear-regression -nl
/simple2_NIDM_examples/datasets.datalad.org/abide/RawDataBIDS/CMU
_a/nidm.ttl,/simple2_NIDM_examples/datasets.datalad.org/abide/Raw
DataBIDS/CMU_b/nidm.ttl -model "fs_000008 = DX_GROUP +
PIQ_tca9ck + http://uri.interlex.org/ilx_0100400" -contrast
DX_GROUP -r L1
```

Figure 2. The full command; an example of a command the user can specify to begin the linear regression on selected variables, with regularization at the end.

In the example shown in [Figure 2](#), we have first run a pynidm query operation on the NIDM documents and identified four variables of interest: supratentorial brain volume (fs_000008), diagnostic group (DX_GROUP), performance IQ (PIQ_tca9ck), and age. The model specified establishes the relationship between the DV, brain volume, and the IVs, diagnostic group, performance IQ, and age. In this example, fs_000008 is the fixed unique identifier (UUID) of the supratentorial brain volume computed with the FreeSurfer software ([Fischl 2012](#)) ([FreeSurfer](#), RRID:SCR_001847) using the original Magnetic Resonance Imaging (MRI) structural scans of the brain. This particular UUID is fixed because it identifies a specific brain region and measurement computed with the FreeSurfer software and will not change across datasets that derive brain volume measurements with FreeSurfer. DX_GROUP is the name of the study-specific variable describing the diagnostic group assigned to participants. PIQ_tca9ck is the performance IQ measure collected on study participants and is the UUID created for this data element when the NIDM documents were created for this dataset. Note, this particular UUID is not guaranteed to be the same across NIDM documents from different studies. Finally, ["http://uri.interlex.org/ilx_0100400"](http://uri.interlex.org/ilx_0100400) is the age of the participants using a URL form to reference a concept describing the high-level measure of age which has been used to annotate the variables measuring age across studies. Here we use a concept URL that has been mapped to each dataset's separate variables that store the age of participants. By using the concept URL, we avoid the complexity of different variable names being used to store consistent information (e.g., age) across datasets.

This example shows that one can select data elements from the NIDM documents for linear regression using three specific forms: (1) using the UUID of the objects in the NIDM graph documents; (2) using the distinct variable name from the original dataset, also stored as metadata in the NIDM graph documents; (3) using a high-level concept that has been associated with specific variables described by the concept across datasets, used to make querying across datasets with different variable names but measuring the same phenomenon easier. We support these three distinct forms of selecting data elements to enable distinct usage patterns. Some investigators will use NIDM documents of their internal studies and want to be able to reference data elements using their study-specific variable names. Other investigators may want to use variables from different studies and thus the variable names are unlikely to be the same; thus, we support the use of selecting variables based on high-level concepts. In practice, users will not often mix forms of referring to data elements within the same model, but we show it here to make evident the flexibility of the tool.

The optional “-contrast” parameter allows one to select one or more IVs to contrast the parameter estimates for those IVs. The contrast variable in this example is “DX_GROUP” which describes the diagnostic group of each participant. Our tool supports multiple methods of coding treatment variables (e.g., treatment coding ([Figure 3](#)), simple coding, sum coding, backward difference coding, and Helmert coding) as made available by the [Patsy Python library](#) ([Brooke 1923](#)). The user can select multiple independent variables to contrast and/or contrasts on interactions. The results of the treatment coding contrast applied in [Figure 2](#) can be seen in [Figure 3](#).

```
Treatment (Dummy) Coding: Dummy coding compares each level of the categorical variable to a base reference level. The base reference level is the value of the intercept.
With contrast (treatment coding)
OLS Regression Results
=====
Dep. Variable:      fs_000008      R-squared:      0.242
Model:              OLS           Adj. R-squared: 0.110
Method:             Least Squares F-statistic:      1.835
Date:               Thu, 08 Jul 2021 Prob (F-statistic): 0.157
Time:               11:40:17       Log-Likelihood: -94.348
No. Observations:   28            AIC:              198.7
Df Residuals:       23            BIC:              205.4
Df Model:           4
Covariance Type:    nonrobust
=====
                    coef    std err          t      P>|t|      [0.025     0.975]
-----
Intercept              8.9220        3.995        2.233     0.036     0.657     17.187
C(DX_GROUP, Treatment)[T.1]  0.5595        3.007        0.186     0.854    -5.660      6.779
C(DX_GROUP, Treatment)[T.2]  9.0486        9.132        0.991     0.332    -9.842     27.940
ilx_0100400            0.8798        0.443        1.987     0.059    -0.036      1.796
PIQ_tca9ck            -0.1204        0.271       -0.445     0.661    -0.681      0.440
=====
Omnibus:              0.227    Durbin-Watson:      1.829
Prob(Omnibus):        0.893    Jarque-Bera (JB):    0.427
Skew:                 -0.067    Prob(JB):            0.808
Kurtosis:              2.410    Cond. No.             78.1
=====
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

Figure 3. Output of command in [Figure 2](#) with treatment coding (contrast using diagnostic group); an example of the printout given after linear regression analysis is complete.


```
pynidm linear-regression -nl ../MTdemog_aseg_v2.ttl
-model "fs_003343 = age*sex + sex + age + group +
age*group" -contrast "age,group" -r L2
```

Figure 4. Model command with interacting variables and multivariable contrast; an example of another command the user can specify to conduct a linear regression analysis with interactions encoded and a multivariable contrast.

Helmert Coding: Our version of Helmert coding is sometimes referred to as Reverse Helmert Coding. The mean of the dependent variable for a level is compared to the mean of the dependent variable over all previous levels. Hence, the name 'reverse' being sometimes applied to differentiate from forward Helmert coding.

OLS Regression Results						
	coef	std err	t	P> t	[0.025	0.975]
Dep. Variable:	fs_003343		R-squared:	0.484		
Model:	OLS		Adj. R-squared:	0.127		
Method:	Least Squares		F-statistic:	1.356		
Date:	Wed, 22 Dec 2021		Prob (F-statistic):	0.299		
Time:	12:36:03		Log-Likelihood:	-156.01		
No. Observations:	23		AIC:	332.0		
Df Residuals:	13		BIC:	343.4		
Df Model:	9					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
Intercept	-59.5015	163.368	-0.364	0.722	-412.437	293.434
C(age + group, Helmert)[H.9]	-62.0343	111.738	-0.555	0.588	-303.430	179.361
C(age + group, Helmert)[H.10]	-95.2003	65.785	-1.447	0.172	-237.321	46.921
C(age + group, Helmert)[H.11]	-122.0933	47.267	-2.583	0.023	-224.208	-19.978
C(age + group, Helmert)[H.12]	-72.9157	48.257	-1.511	0.155	-177.169	31.338
C(age + group, Helmert)[H.13]	-300.9541	116.603	-2.581	0.023	-552.859	-49.049
age	388.2647	12.304	31.555	0.000	361.683	414.847
group	-1339.6296	1773.222	-0.755	0.463	-5170.443	2491.184
age:group	202.8076	200.736	1.010	0.331	-230.857	636.472
sex	2043.2521	843.190	2.423	0.031	221.651	3864.853
age:sex	-197.9404	84.993	-2.329	0.037	-381.557	-14.323
Omnibus:	1.974	Durbin-Watson:	2.337			
Prob(Omnibus):	0.373	Jarque-Bera (JB):	1.161			
Skew:	0.207	Prob(JB):	0.560			
Kurtosis:	1.980	Cond. No.	1.14e+17			

Figure 5. Output of command in Figure 4 with Helmert coding (contrast using age and group); an example of the printout given after linear regression analysis is complete.

The optional “-r” parameter allows the user to select L1 (Lasso) or L2 (Ridge) regularization implemented in scikit-learn (Varoquaux *et al.* 2015) (scikit-learn, RRID:SCR_002577). In either case, regularizing prevents the data from being overfit, potentially improving model generalizability and demonstrating which variables have the strongest relationships with the dependent variable. The regularization weight is iteratively determined across a wide range of regularization weightings using 10-fold cross-validation, selecting the regularization weight yielding the maximum likelihood.

There are error checks within the code to make sure the researcher has feedback on why a model cannot run, whether it is because there are not enough data points or because one or more variables could not be found in one or more of the NIDM documents. This makes the experience as simple as possible for the user, which is important, as our intended audience for these tools are investigators who may have no prior experience with the semantic web and/or NIDM documents.

In the example shown in Figure 4, we have first run a pynidm query operation on the NIDM documents and identified 4 variables of interest: fs_003343, age, sex, and group. Here, fs_003343 is the fixed unique identifier (UUID) of the left hippocampus volume while age, sex, and group are names of the study-specific variables concerning the age of the participant at the time of the study, the gender, and the group the participant was in. The model specified establishes the relationship between the DV, left hippocampus volume, and the IVs, group, age, and sex. However, in this case, we also encode interactions between age and sex and age and group, as denoted by the asterisks. Also, in this model, we have used multiple IVs to contrast the parameter estimates for those IVs. The contrast variables are age and group. Finally, L2 regularization is selected for regularization.

The results of the Helmert coding contrast can be seen in Figure 5.

Operation

The data must be in a NIDM document(s) to be used with this tool. Data can be transformed into a NIDM document directly from BIDS or tabular data files using the PyNIDM tools “bidsmri2nidm” and “csv2nidm”. Once data is transformed into NIDM, the user only needs to have a functional installation of PyNIDM and access to a terminal window or similar command-line processing tool with a functional version of Python 3. Once the user specifies the parameters, data is aggregated from the NIDM files, re-structured for the linear regression algorithm, and parameter

estimates learned using ordinary least squares and returning either a printout or output file of the various coefficients and summary statistics.

Conclusions

In this work, we have designed a linear regression tool that works on linked-data NIDM documents in support of understanding relationships between variables collected across studies. This tool helps scientists evaluate relationships between data prior to fully integrating datasets for hypothesis testing which may require considerable time and resources. In our initial evaluations, this tool has shown utility for these use cases. In future work, we are creating additional machine learning tools allowing users to cluster data in a similar fashion to the linear regression tool presented here. Further, the NIDM community is working on additional functionality for the PyNIDM toolkit that transforms the value representations of the variables selected for modeling to be consistent across all NIDM documents used in the model. These transformations are made using the detailed data dictionaries included in the NIDM documents. This functionality will be included in the PyNIDM query application programmers' interface (API) and will be immediately available to the linear regression tool presented here.

Software availability

Source code available from: https://github.com/incf-nidash/PyNIDM/blob/master/nidm/experiment/tools/nidm_linreg.py

Archived source code at time of publication: v3.9.5, <https://doi.org/10.5281/zenodo.4635287> (Keator *et al.*, 2021)

License: [Apache License, Version 2.0](#)

References

- Apache License, Version 2.0: **Copyright 2022, The Apache Software Foundation.**
[Reference Source](#)
- Brooke MC: **Patsy. *The Psychological Clinic* 1923; 15(1-2): 41–43.**
[PubMed Abstract](#)
- Consortium for Reliability and Reproducibility (CoRR) — Consortium for Reliability and Reproducibility (CoRR) Documentation.
[Reference Source](#)
- Di Martino A, Yan C-G, Li Q, *et al.*: **The Autism Brain Imaging Data Exchange: Towards a Large-Scale Evaluation of the Intrinsic Brain Architecture in Autism. *Molecular Psychiatry* 2014; 19(6): 659–667.**
[PubMed Abstract](#) | [Publisher Full Text](#)
- Fischl B: **FreeSurfer. *NeuroImage* 2012; 62: 774–781.**
[PubMed Abstract](#) | [Publisher Full Text](#)
- Gorgolewski K, Auer T, Calhoun V, *et al.*: **The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments. *Scientific Data* 2016; 3: 160044.**
[PubMed Abstract](#) | [Publisher Full Text](#)
- Keator D: **simple2_NIDM_examples: ReproNim simple2 NIDM Files for ABIDE/ADHD200 Available from Datalad. Github.**
[Reference Source](#)
- Keator D, *et al.*: **incf-nidash/PyNIDM: PyNIDM tools v3.7.6 (dev) (v3.7.6). Zenodo 2021.**
[Publisher Full Text](#)
- Keator DB, Helmer K, Steffener J, *et al.*: **Towards Structured Sharing of Raw and Derived Neuroimaging Data across Existing Resources. *NeuroImage* 2013; 82(November): 647–661.**
[PubMed Abstract](#) | [Publisher Full Text](#)
- Maumet C, Auer T, Bowring A, *et al.*: **Sharing Brain Mapping Statistical Results with the Neuroimaging Data Model. *Scientific Data* 2016; 3(December): 160102.**
[PubMed Abstract](#) | [Publisher Full Text](#)
- Milham MP, Buitelaar J, Castellanos FX, *et al.*: **ADHD200. 2011.**
[Reference Source](#)
- Moreau L, Ludäscher B, Altintas I, *et al.*: **Special Issue: The First Provenance Challenge. *Concurrency and Computation: Practice & Experience* 2008; 20(5): 409–418.**
[Publisher Full Text](#)
- Nagpal A: **L1 and L2 Regularization Methods. *Towards Data Science* 2017. October 13, 2017.**
[Reference Source](#)
- nidm_linreg: **Github. Copyright 2017-202, INCF-NIDASH developers.**
[Reference Source](#)
- NIDM Working Group: **NIDM. Copyright 2018, NIDM Working Group.**
[Reference Source](#)
- OpenNeuro: **Copyright 2022, Stanford Center for Reproducible Neuroscience.**
[Reference Source](#)
- PROV-Overview: **Copyright 2013, W3C.**
[Reference Source](#)
- PyNIDM: **Github. Copyright 2017-2020, INCF-NIDASH developers.**
[Reference Source](#)
- Ravan J, Person DY, Packer J, *et al.*: **What Is REST. 2020. May 31, 2020.**
[Reference Source](#)
- RDF 1.1 Turtle: **Copyright 2008-2014, W3C.**
[Reference Source](#)
- Ripley BD: **The R Project in Statistical Computing. *MSOR Connections* 2001; 1(1): 23–25.**
[Publisher Full Text](#)
- Semantic Web - W3C: **Copyright 2015, W3C.**
[Reference Source](#)
- SPARQL Query Language for RDF: **Copyright 2006-2007, W3C.**
[Reference Source](#)
- Varoquaux G, Buitinck L, Louppe G, *et al.*: **Scikit-Learn. *GetMobile Mobile Computing and Communications* 2015; 19(1): 29–33.**
[Publisher Full Text](#)
- Welcome to Click — Click Documentation (8.0.X): **Copyright 2014, Pallets.**
[Reference Source](#)
- Zuo X-N, Anderson JS, Bellec P, *et al.*: **An Open Science Resource for Establishing Reliability and Reproducibility in Functional Connectomics. *Scientific Data* 2014; 1(December): 140049.**
[PubMed Abstract](#) | [Publisher Full Text](#)

Open Peer Review

Current Peer Review Status:

Version 1

Reviewer Report 11 April 2022

<https://doi.org/10.5256/f1000research.119281.r125519>

© 2022 Specht K. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Karsten Specht

Department of Biological and Medical Psychology, University of Bergen, Bergen, Norway

In this software tool article, the authors describe a tool that allows high-level statistical analysis for combining studies that follow the Neuroimaging Data Model (NIDM) framework. The tool is/will be part of the PyNDIM toolbox.

In general, this appears as a very useful tool that a larger community might want to use. However, the article is phrased and organized in a way that this might not happen. The audience of the current manuscripts is almost exclusively the NDIM community that is involved in developing the tools, but it is not directed to potential users. The main problem is, in my view, that the level of explanation and description of the tool is very minimalistic. Of course, all important aspects and relevant tools are mentioned, but only as keywords with the appropriate references. My main concern is that readers not entirely familiar with all aspects and methods incorporated into this tool will not get its importance and relevance.

I would like to see just a few more sentences describing the key elements in a revised version. For example, readers unfamiliar with the PROV standard or the semantic web technique are already lost after the third sentence. And the introduction continues like that. For example, the “L1 and L2 regularization” is just thrown in on the first page, assuming that every reader immediately gets its relevance, but a more relevant discussion follows much later.

Similarly, I would have expected a more detailed description of the implementations in the section that is called “implementation and use cases”, but this section is mainly on “use cases” and very little on implementation.

So, in summary, I think a revised version of this article would benefit from just a few more describing lines so that this very relevant tool generates interest in a broader audience.

Is the rationale for developing the new software tool clearly explained?

Yes

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Partly

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Neuroimaging, methods development, reliability measures

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 30 Apr 2022

Ashmita Kumar

Thank you for the time you have taken to review this paper. We will add more specifics to make it more accessible to potential users, defining our implementation, data structures, and technique in more detail. We apologize for the time it has taken to respond, but rest assured we are working to address it in our next version.

Competing Interests: No competing interests were disclosed.

Reviewer Report 08 April 2022

<https://doi.org/10.5256/f1000research.119281.r125517>

© 2022 Thomas A et al. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. The author(s) is/are employees of the US Government and therefore domestic copyright protection in USA does not apply to this work. The work may be protected under the copyright laws of other jurisdictions when used in those jurisdictions.

Adam G. Thomas

Data Sharing and Science Team, National Institute of Mental Health, Bethesda, MD, USA

Eric Earl

Data Sharing and Science Team, National Institute of Mental Health, Bethesda, MD, USA

The authors have written about the `nidm_linreg` tool which is a linear regression tool that is now a part of the PyNIDM software. `nidm_linreg` was added because PyNIDM did not have a tool like this in its suite yet. The tool's design and the paper's focus on usage are both very thorough. This software tool article is likely of interest to PyNIDM and other members of the neuroimaging community. We have the following comments and concerns about the software and the paper:

1. Echoing the concern expressed in Dec 2021 in the PyNIDM GitHub issue 309 (<https://github.com/incf-nidash/PyNIDM/issues/309>), it is bad practice to install Python packages outside of dependency management functionality. The `nidm_linreg.py` tool currently tries to quietly install 4 packages using `pip` if they are not present in the Python environment. This will likely cause problems and/or confusion for the user. A safe and reproducible way to provide the end-user with the dependencies they need is to provide a `requirements.txt` file with the repository which lets users use a specific version of the expected package dependencies. <https://pip.pypa.io/en/stable/reference/requirements-file-format/>
2. We ran the commands on the GitHub README.md (with some necessary modifications) to install and test PyNIDM (see below). `pytest` returned several errors and warnings (on two different platforms) that should be corrected:


```
conda create -n pynidm_py3 python=3
git clone https://github.com/incf-nidash/PyNIDM.git
cd PyNIDM
pip install -e .
pytest
[snip]
= 4 failed, 24 passed, 14 skipped, 3 warnings, 36 errors in 35.41s ==
```
3. Running the commands as-written in Figures 2 and 4 did not work. We cloned the `simple2_NIDM_examples` repository (https://github.com/dbkeator/simple2_NIDM_examples) and tried to use the data within, but we were only able to run one of the example commands successfully.

This felt insufficient to evaluate the reproducibility of the tool. The authors should provide commands, script, or a container such that users can reproduce the same outputs presented in the paper.

4. The Methods section "Implementation and use cases" covers a blend of those two things where we think it would read better to see a focused section on Implementation and a focused section on Use cases. The Implementation section could discuss the design decisions and software architecture as well as planned support. Most of the current content could move over to a Use cases section covering a use case or two (as it does now).
5. The paper mentioned there are "error checks" built into the tool, but no examples are

shown. Including a sentence about the kinds of errors being checked for would be another useful addition.

6. The reference for the Patsy Python Library (Brooke 1923).on page 5 of the PDF (Methods section, 8th paragraph) is incorrect.

Is the rationale for developing the new software tool clearly explained?

Yes

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

No

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

Partly

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Neuroimaging, Neuroscience, Computer Science, Data Science

We confirm that we have read this submission and believe that we have an appropriate level of expertise to state that we do not consider it to be of an acceptable scientific standard, for reasons outlined above.

Author Response 30 Apr 2022

Ashmita Kumar

Thank you for your feedback on our tool. We appreciate your thoroughness and will ensure to make the required changes. To your first point, we will make sure that our package requirements are presented to the user before installation instead of using a quiet install. To your second point, while we are unsure why the error appeared, we are looking into it and attempting to reproduce it. We will be sure to address it in our next version. To your third point, we will make sure to provide a way for users to replicate the outputs in the paper by providing more specifics on how we obtained them. To your fourth point, we will separate the two sections and add to our Implementation section in our next version. To your fifth point, the error checks built in include data validation for type and quantity of that data, which we will explain in more detail. To your final point, we will try to find the correct citation.

Thank you for taking the time to make this review, and we are sorry for the length of time it has taken to respond. We really appreciate it.

Competing Interests: No competing interests were disclosed.

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com

F1000Research