WHITE PAPER

SDRF (Scientific Data Repository Framework): Improving Data Sets' Machine Readabability and Interoperability with Published Research

A New Protocol for Sharing Scientific Data

Linguistic Technology Systems (LTS) is developing **SDRf** to encompasses data models, publishing guidelines, and code libraries to for deploying open-access research data sets associated with scientific publications. Nowadays there are many general-purpose and domain-specific portals hosting scientific data; there are also several available formats for describing and encoding scientific data, such as Research Objects, **schema.org/Dataset**, Digital Curation Center (**DCC**), **SciData**, **BioCoder**, and **MIBBI** (Minimum Information for Biological and Biomedical Investigations). However, **SDRf** is unique in merging different data-set formats into a unified, overarching standard which can be adapted to different publishing housing and pipelines.

Shaping Protocols to Conform to Current Specifications in Publishing

In order to conform to current specifications — such as FAIRsharing (Findable, Accessible, Interoperable, Reusable) or the Bill and Melinda Gates Foundation guidelines for authors (https://gatesopenresearch.org/for-authors/data-guidelines) — proper protocols must be implemented at several stages of the publishing process: (1) publications should provide clear descriptions of accompanying data sets and where that data is hosted; (2) publication repositories should make data-set links and metadata clearly visible on web pages where documents are read or previewed; (3) data sets themselves need to include metadata and supporting files which help researchers properly access, visualize, and reuse the data; and (4) data sets need to be connected with software which has the correct features to load and display the relevant raw data files. **SDRf** will include technology applicable to each of these four facets of the publishing and data-sharing pipeline in order to conform to current standards.

It is important to emphasize that data sets are only truly valuable if they are machine-readable and seamlessly integrated into domain-specific software ecosystems. Scientists who examine and reuse published data sets are generally researchers doing technical work in a field closely related to that of the original authors; in many cases there are specialized software applications, computational methods, algorithms, and research protocols which are endemic to the relevant subject areas. When sharing research data, accordingly, publishers/authors should make it as easy as possible for scientists to examine the data within the digital ecosystem that they utilize for their own research. This often implies that document viewers — e.g., **PDF** viewers and/or **HTML** pages on publisher portals — should ideally be interconnected with scientific applications so that scientists, when reading books/articles, can seamlessly launch domain-specific software and visualize/examine associated data sets. Unfortunately, most scientific software does not incorporate code libraries to parse metadata (summarizing file types, download instructions, etc.) describing open-access data sets. This can be addressed by providing plugins or extensions that add data-set-accession capabilities to existing scientific software, so that publisher repositories and data-hosting repositories can be made truly interoperable with scientific applications.

To demonstrate how such plugins can work, as well as other facets of the data-publication process facilitated via **SDRf**, this paper will review two case-studies in the academic literature.

First Case Study: "Parkinson's Disease Diagnosis: Effect of Autoencoders to Extract Features from Vocal Characteristics," by Ashena Gorgan Mohammadi, Pouya Mehralian, Amir Naseri, and Hedieh Sajedi (International Journal of Speech Technology, pending review)

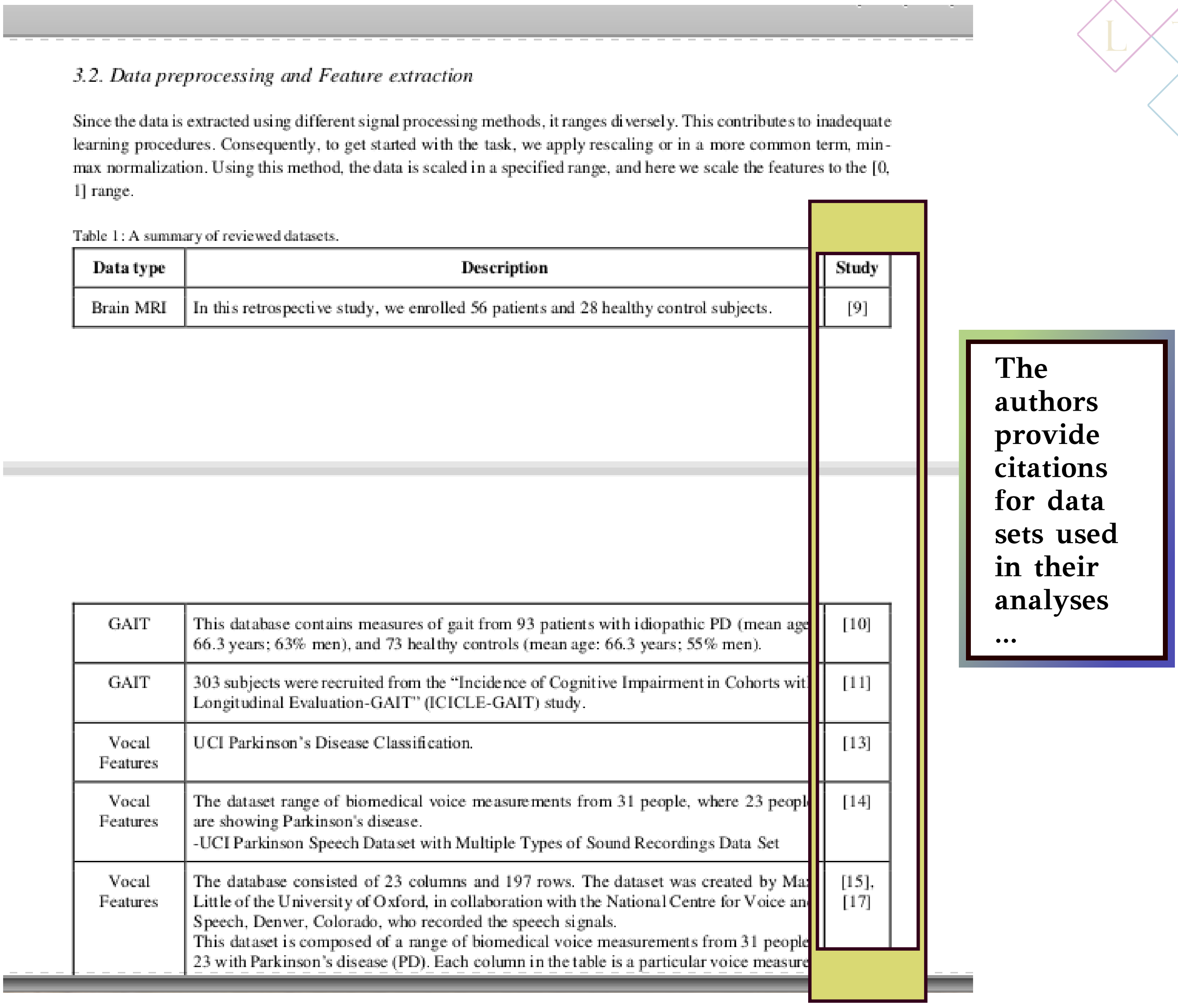
This case study demonstrates a scenario where an article reuses multiple pre-existing data sets. The article examines Parkinson's Disease symptomology from multiple perspectives, including gait (loss of motor function), speech impairment, and bioimaging (MRIs). The authors apply Machine Learning to data sets focused on these three different diagnostic areas, in an effort to advance research to refine Parkinson's predictors and diagnoses. The overall information can be summarized as follows:

1. **Gait Data**: this data was primarily drawn from a PhysioNet data set (https://physionet.org/content/gaitpdb/1.0.0/) obtained via sensors attached to subject's feet as they walked (the study includes both Parkinson's patients and healthy controls). This sensor data is provided as a collection of text files, each file corresponding to one patient (or control subject), with each line in a file representing a single time snapshot. The lines are divided into space-separated columns, each representing force exerted on a single sensor, plus two additional columns calculating total force on the left-foot and right-foot sensors respectively. This data set also includes demographic and clinical information for each patient in a spreadsheet format. The authors also use an additional source of gait data derived from a more recent (2019) study whose data is available only upon request (see https://www.nature.com/articles/s41598-019-53656-7#MOESM1).
2. **Speech Data**: this data was primarily drawn from a data set hosted by the University of California Irvine Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/Parkinsons). The central information is a CSV file, where each line represents a single voice recording from one of 31 subjects (consisting of 23 Parkinson's patients and 8 healthy controls). Each subject made multiple recordings. The individual lines in the CSV file present a quantitative model of subjects' speech via a collection of acoustic features/attributes. A similar "telemonitoring" data set from the same archive was used to study how the analysis of Parkinson's-related speech data may be applicable to samples obtained via devices such as smartphones.
3. **Radiological Data**: this data is not immediately available for reuse, but requires special (non-commercial) authorization from the Parkinson's Progressive Markers Initiative (https://www.ppmi-info.org/) or a request made of corresponding authors of referenced papers introducing the relevant data (https://www.frontiersin.org/articles/10.3389/fnins.2019.00874/full#h6). [1] Although this data is derived from bioimaging (so that the underlying raw data files are radiological) the authors of this paper under review utilize the data in a more structured form, building off of image-feature extraction already performed when the data sets were first published. However, the republished unified data set itself (to appear in conjunction with the IJST submission) might include code to allow researchers to reproduce the original analytic workflow if desired: for instance, among other things, we can enable the implementations published via https://biomedia.doc.ic.ac.uk/software/malp-em/ to be embedded as a CaPTk module (see in particular https://cbica.github.io/CaPTk/tr_integration.html#tr_cppIntegration).
4. **Python Code Repository**: the authors also provide Python code, hosted on GitHub, which they used to analyze these various data sources.

Uniying the Data Sets into a Single Package

In this artice pending review, the authors summarize the data sets in a table within the main text (see Figure 1 here) and, in their bibliography, they cite these data sets either directly or by referencing publications where the relevant data sets are described (see Figure 2 here). In so doing, this properly credits authorship to the researchers who curated the data sets, and it gives readers a means to locate the raw data. However, accessing and working with the raw data is inconvenient from a reader's point of view without most or all of the data sets being repackaged into a *single* archive that could be hosted and downloaded as one unit. Obtaining raw data from the resources identified in the

[1] When the unified data set is published, we will request permission to include a copy of the original data set to spare future readers having to request this data on their own



3.2. Data preprocessing and Feature extraction

Since the data is extracted using different signal processing methods, it ranges diversely. This contributes to inadequate learning procedures. Consequently, to get started with the task, we apply rescaling or in a more common term, min-max normalization. Using this method, the data is scaled in a specified range, and here we scale the features to the [0, 1] range.

Table 1 : A summary of reviewed datasets.

Data type	Description	Study
Brain MRI	In this retrospective study, we enrolled 56 patients and 28 healthy control subjects.	[9]
GAIT	This database contains measures of gait from 93 patients with idiopathic PD (mean age 66.3 years; 63% men), and 73 healthy controls (mean age: 66.3 years; 55% men).	[10]
GAIT	303 subjects were recruited from the "Incidence of Cognitive Impairment in Cohorts wit Longitudinal Evaluation-GAIT" (ICICLE-GAIT) study.	[11]
Vocal Features	UCI Parkinson's Disease Classification.	[13]
Vocal Features	The dataset range of biomedical voice measurements from 31 people, where 23 peopl are showing Parkinson's disease. -UCI Parkinson Speech Dataset with Multiple Types of Sound Recordings Data Set	[14]
Vocal Features	The database consisted of 23 columns and 197 rows. The dataset was created by Ma Little of the University of Oxford, in collaboration with the National Centre for Voice an Speech, Denver, Colorado, who recorded the speech signals. This dataset is composed of a range of biomedical voice measurements from 31 people 23 with Parkinson's disease (PD). Each column in the table is a particular voice measure	[15], [17]

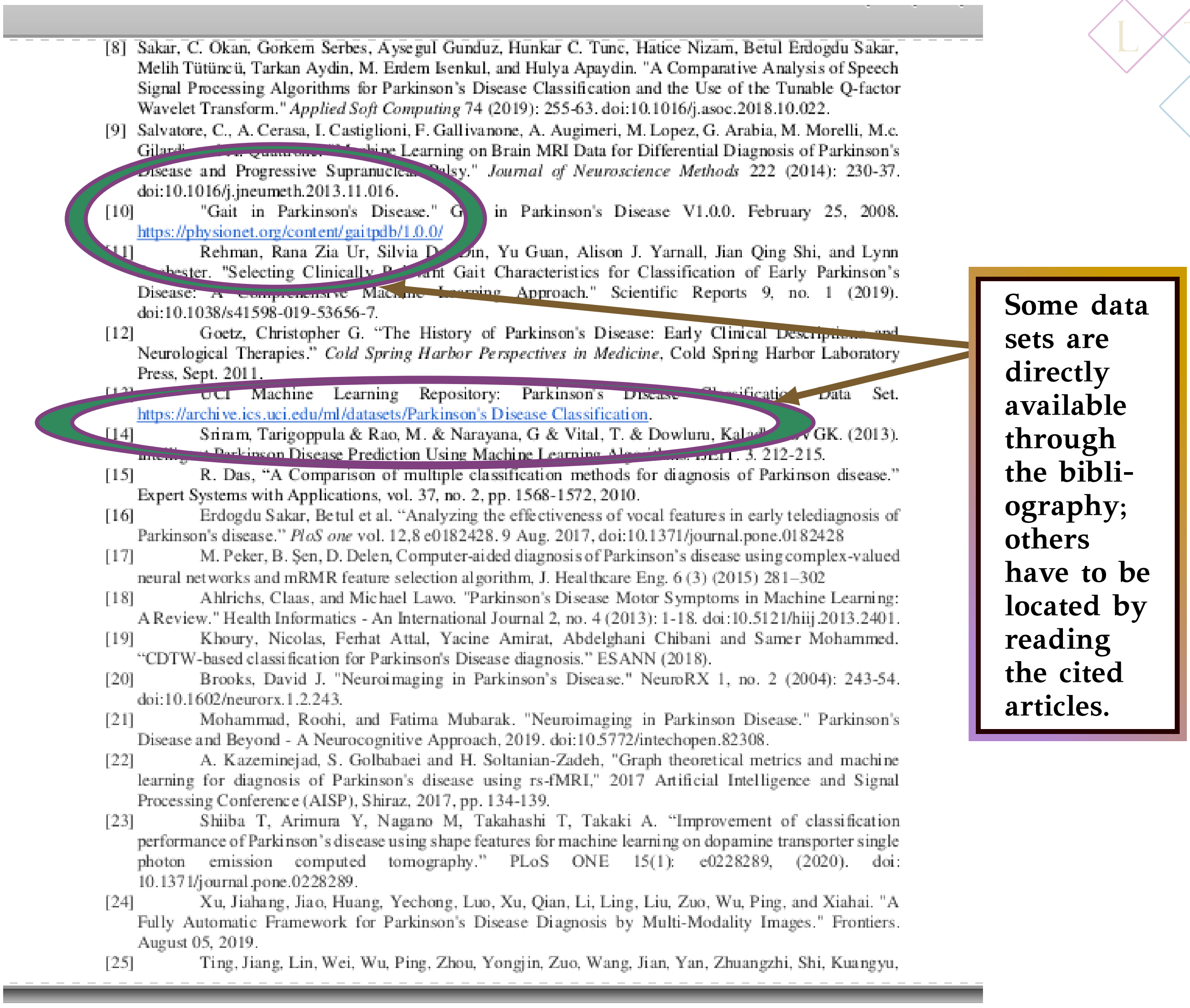
Figure 1: Table Listing Analyzed Data Sets in the Parkinson's Article

bibliography requires several steps — for instance, the PhysioNet sensor data can be downloaded as a zipped folder, while the demographic data attached to it has to be downloaded separately. Furthermore, some of the information obtained from **MRI** and speech analysis (reported in papers cited as data sources for the IJST submission) is provided as supplemental materials within the secondary papers; this requires readers to browse one at a time through each of the relevant articles so as to find downloadable links (see Figure 4). In short, piecing all the source data together puts the onus on readers to manually inspect multiple web resources and to manually interconnect files once they are downloaded.

Another complicating factor is that certain information present in the data sets is implicit within how the data sets are organized, requiring extra effort to extract this information in a machine-readable manner. For instance, the PhysioNet sensor data uses a file-naming convention which encodes several pieces of information in the file names, such as whether the file presents a Parkinson's patient or a control subject (see Figure 3). Though by examining file names it would be possible to construct a table with additional information providing context for the file contents, such information is not directly included within the PhysioNet data set; it needs to be extracted by computer code.

Constructing Machine-Readable Supplemental Archives

This collection of data sets serves as an example of how technologies such as **SDRf** can fill the gap between publication/data repositories and scientific computing, making scientific data more "**FAIR**" (Findable, Accessible, Interoperable, Reusable). Upon publication of this submission in IJST (pending peer review) the disparate open-access data from the article's secondary sources would be provided as a single **SDRf** archive. This archive would provide *machine-readable* access to information spread across multiple sources, translated into a common file format. In general, **SDRf** encourages and implements features to help data sets conform to **FAIR** and related standards, such as (1) bundling multiple data sets into a single archive; (2) migrating data to general-purpose representations wher-



[8] Sakar, C. Okan, Gorkem Serbes, Aysegul Gunduz, Hunkar C. Tunc, Hatice Nizam, Betul Erdogdu Sakar, Melih Tütüncü, Tarkan Aydin, M. Erdem Isenkul, and Hulya Apaydin. "A Comparative Analysis of Speech Signal Processing Algorithms for Parkinson's Disease Classification and the Use of the Tunable Q-factor Wavelet Transform." *Applied Soft Computing* 74 (2019): 255-63. doi:10.1016/j.asoc.2018.10.022.

[9] Salvatore, C., A. Cerasa, I. Castiglioni, F. Gallivanone, A. Augimeri, M. Lopez, G. Arabia, M. Morelli, M.c. Gilar[illegible] Learning on Brain MRI Data for Differential Diagnosis of Parkinson's [illegible]isease and Progressive Supranucle[illegible]lsy." *Journal of Neuroscience Methods* 222 (2014): 230-37. doi:10.1016/j.jneumeth.2013.11.016.

[10] "Gait in Parkinson's Disease." G[illegible] in Parkinson's Disease V1.0.0. February 25, 2008. https://physionet.org/content/gaitpdb/1.0.0/

[11] Rehman, Rana Zia Ur, Silvia D[illegible]in, Yu Guan, Alison J. Yarnall, Jian Qing Shi, and Lynn [illegible]chester. "Selecting Clinically R[illegible]nt Gait Characteristics for Classification of Early Parkinson's Disease: A [illegible] Machine [illegible] Approach." Scientific Reports 9, no. 1 (2019). doi:10.1038/s41598-019-53656-7.

[12] Goetz, Christopher G. "The History of Parkinson's Disease: Early Clinical Descrip[illegible] and Neurological Therapies." *Cold Spring Harbor Perspectives in Medicine*, Cold Spring Harbor Laboratory Press, Sept. 2011.

[13] UCI Machine Learning Repository: Parkinson's Disease Classificatio[illegible] Data Set. https://archive.ics.uci.edu/ml/datasets/Parkinson's Disease Classification.

[14] Sriram, Tarigoppula & Rao, M. & Narayana, G & Vital, T. & Dowluru, Kal[illegible]GK. (2013). [illegible] Parkinson Disease Prediction Using Machine Learning [illegible]. 3. 212-215.

[15] R. Das, "A Comparison of multiple classification methods for diagnosis of Parkinson disease." Expert Systems with Applications, vol. 37, no. 2, pp. 1568-1572, 2010.

[16] Erdogdu Sakar, Betul et al. "Analyzing the effectiveness of vocal features in early telediagnosis of Parkinson's disease." *PloS one* vol. 12,8 e0182428. 9 Aug. 2017, doi:10.1371/journal.pone.0182428

[17] M. Peker, B. Şen, D. Delen, Computer-aided diagnosis of Parkinson's disease using complex-valued neural networks and mRMR feature selection algorithm, J. Healthcare Eng. 6 (3) (2015) 281–302

[18] Ahlrichs, Claas, and Michael Lawo. "Parkinson's Disease Motor Symptoms in Machine Learning: A Review." Health Informatics - An International Journal 2, no. 4 (2013): 1-18. doi:10.5121/hiij.2013.2401.

[19] Khoury, Nicolas, Ferhat Attal, Yacine Amirat, Abdelghani Chibani and Samer Mohammed. "CDTW-based classification for Parkinson's Disease diagnosis." ESANN (2018).

[20] Brooks, David J. "Neuroimaging in Parkinson's Disease." NeuroRX 1, no. 2 (2004): 243-54. doi:10.1602/neurorx.1.2.243.

[21] Mohammad, Roohi, and Fatima Mubarak. "Neuroimaging in Parkinson Disease." Parkinson's Disease and Beyond - A Neurocognitive Approach, 2019. doi:10.5772/intechopen.82308.

[22] A. Kazeminejad, S. Golbabaei and H. Soltanian-Zadeh, "Graph theoretical metrics and machine learning for diagnosis of Parkinson's disease using rs-fMRI," 2017 Artificial Intelligence and Signal Processing Conference (AISP), Shiraz, 2017, pp. 134-139.

[23] Shiiba T, Arimura Y, Nagano M, Takahashi T, Takaki A. "Improvement of classification performance of Parkinson's disease using shape features for machine learning on dopamine transporter single photon emission computed tomography." PLoS ONE 15(1): e0228289, (2020). doi: 10.1371/journal.pone.0228289.

[24] Xu, Jiahang, Jiao, Huang, Yechong, Luo, Xu, Qian, Li, Ling, Liu, Zuo, Wu, Ping, and Xiahai. "A Fully Automatic Framework for Parkinson's Disease Diagnosis by Multi-Modality Images." Frontiers. August 05, 2019.

[25] Ting, Jiang, Lin, Wei, Wu, Ping, Zhou, Yongjin, Zuo, Wang, Jian, Yan, Zhuangzhi, Shi, Kuangyu,

Figure 2: Bibliography (With Data Set Hyperrefs) in the Parkinson's Article

ever possible — formats such as **XML**, **HDF5**, **ARFF**, or **DICOM**; (3) providing meta-data in multiple formats (**DCC**, **schema.org/Dataset**, **MIBBI**, **BioCoder**, etc.) to be compatible with different organizations' platforms; (4) identifying one or more "preferred applications" for examining/reusing the published data; (5) explicitly representing information encoded via file-names; (6) bundling raw data, meta-data, and (where possible) machine-readable article text into a single resource, which **SDRf** calls a "Supplemental Archive;" and (7) annnotating the data sets to support microcitations that granularly link the publication to its associated Supplemental Archive.

Once a Supplemental Archive has been downloaded, an important question for any **SDRf** archive is how researchers will productively access the data. Unlike the Flow Cytometry use-case discussed below, the Parkinson's archive spans several scientific disciplines; as such, there is no obvious application which could be preferred by default for examining the data files. As a fallback option, **SDRf** is designed to present data sets via **Qt** Creator, a **C++** Integrated Development Environment associated with the **Qt** application-development framework. **SDRf** includes code libraries to represent research meta-data as **C++** objects; these libraries can be opened as **Qt** projects. These may be supplemented with separate libraries extracting and managing information specific to individual data sets. In particular, the Supplemental Archive for the Parkinson's article under peer review would provide **C++** classes encapsulating spreadsheet-like data (whether originally in **.xls**, **CSV**, or space-delimited formats) republished by the Journal in the unified data set.

An additional concern for **SDRf** archives is how to properly annotate publications and data sets side-by-side. In the Parkinson's article, individual **C++** classes encapsulating tabular data serve as convenient microcitation targets: annotations within the relevant **C++** code represent anchors through which the data set may be referenced (on a more precise scale than merely citing the Supplemental Archive as a whole). In some places, individual class attributes can also be linked to lines in the authors' Python source code. On the text side, certain paragraphs within the Parkinson's article can be linked to the corresponding **C++** code annotations. This illustrates **SDRf**'s recommended

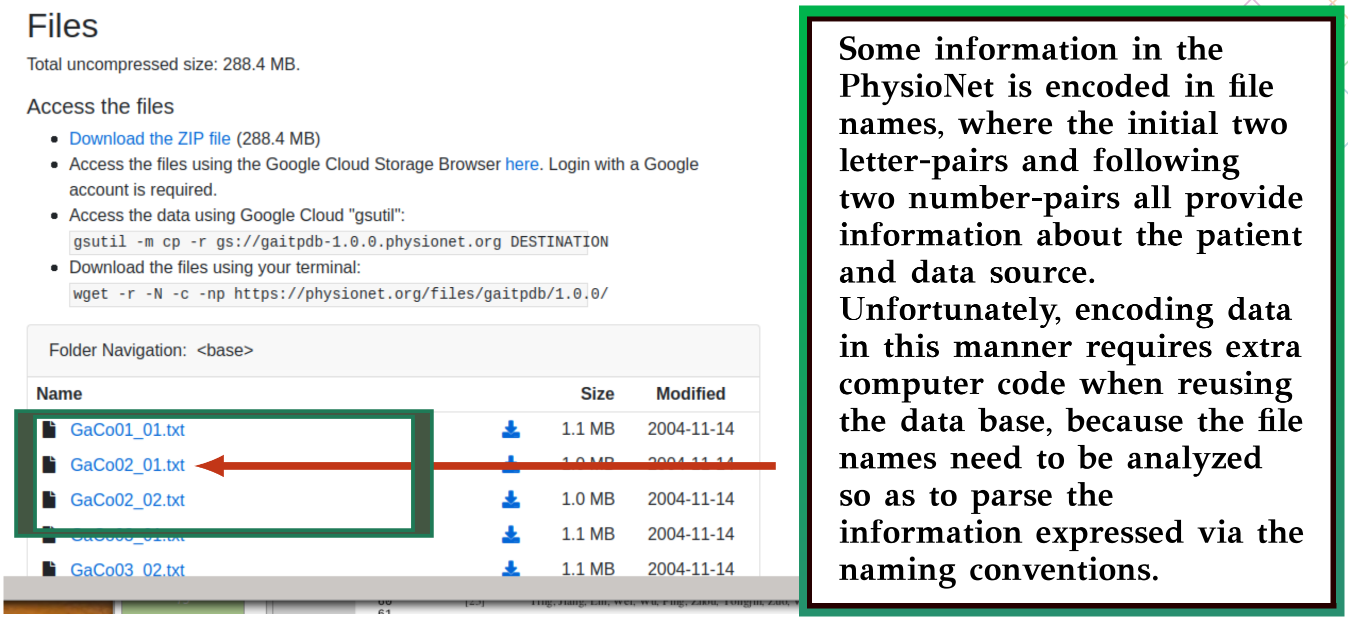


Figure 3: Extracting Information Encoded in File Names

annotation/microcitation system, where segments in publication texts (identified for instance via LaTeX **phantomsection** commands or **JATS statement** tags) are linked to annotations or comments in code and/or raw data files in the Supplemental Archive.

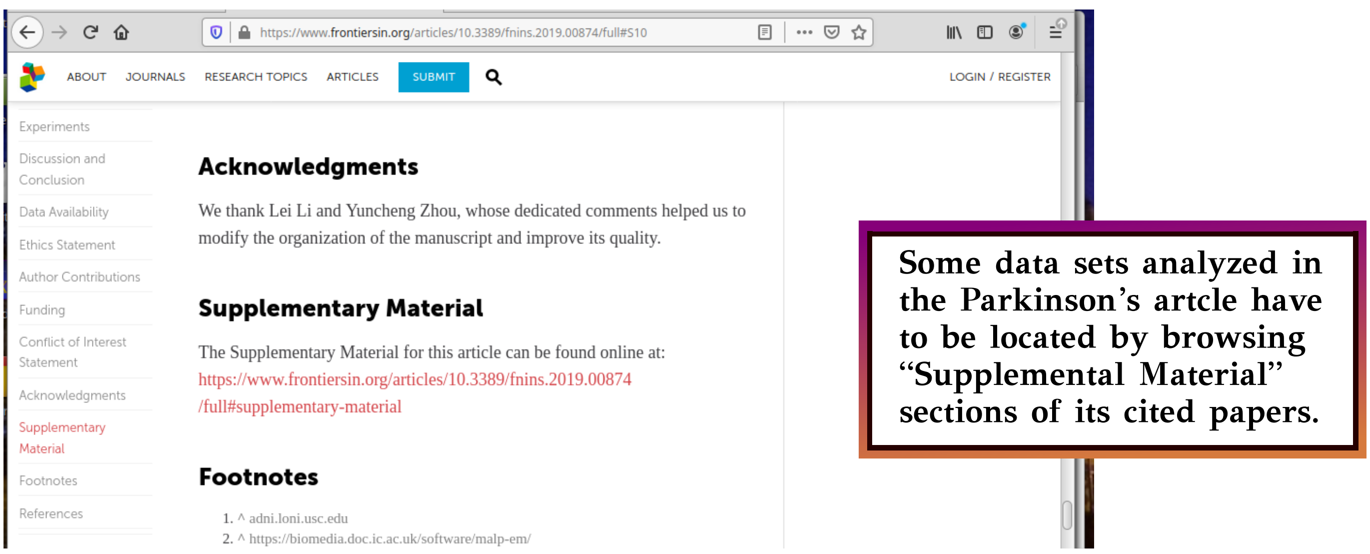


Figure 4: Indirectly Locating Data Sets from Cited Papers

Second Case Study: "Marked T cell activation, senescence, exhaustion and skewing towards TH17 in patients with COVID-19 pneumonia" from nature.com, 2020

This article presents a use-case with some noteworthy contrasts to the Parkinson's publication described in the previous section. This Covid-19 paper (https://www.nature.com/articles/s41467-020-17292-4) was published along with two data sets comprising Flow Cytometry Standard (**FCS**) files hosted via the Flow Repository (http://flowrepository.org/id/FR-FCM-Z2N4 and http://flowrepository.org/id/FR-FCM-Z2N5). Links to the data sets (via **flowrepository.org** pages) are explictly provided in the publication's "Data Availability" section. However, researchers still need to perform several steps to manually download the full set of relevant **FCS** and meta-data files.

One feature of this second use-case is that the technical information in the data sets belong to a single scientific area (Flow Cytometry) and are mostly encoded in a single format (**FCS**). As such, it is straightforward to identify the kind of software which researchers need to use to visualize the raw data — basically, any application that can parse **FCS** files. There are a variety of commercial as well as open-source Flow Cytometry (**FCM**) applications which can be used to access **FCS** data. Once readers have downloaded the Flow Repository archives, they may individually load the **.fcs** files to study data which, in the original article, is summarized via figure illustrations.

This workflow nonetheless requires researchers to perform several manual steps before being able to use the published data. The core problem is that existing Flow Cytometry software does not intrinsically have capabilities to read PDF files, locate FCS data sets, and interoperate with hosting platforms such as Flow Repository. Employing this use-case as an example with which to illustrate proper alignment between document viewers, publication/data repositories, and scientific software, We are developing a new Flow Cytometry application which *can* interoperate with PDF viewers and SDRf archives. This application is designed so that, when authors are reading a PDF file associated with an FCS data set, the PDF viewer can automatically launch and signal to the FCM application when a reader wishes to download and visualize FCS files. In short, the FCM application — having received data from a PDF viewer which implements an SDRf inter-application protocol — will automatically execute download and extraction steps that scientists otherwise would have to perform manually.

Note also that, although most of the relevant data for this Covid-19 article is in FCS form, there is, in addition, supplemental clinical information provided in other formats (...). For cases such as these, the LTS FCM software includes code libraries allowing researchers to parse non-FCS data in standard formats such as XML, HDF5, or ARFF.

This use-case illustrates a general principle: that research data is most convenient for scientists when it is deployed within an infrastructure where portals, document viewers, and scientific applications can seamlessly interoperate. Wherever possible, when researchers are reading published books or articles, they should be able to *automatically* launch the proper scientific application, download data sets, and examine raw data files in the preferred software with only one or two clicks. These steps would be performed automatically as much as is feasible, instead of readers having to waste time with manually finding, downloading, merging/extracting, and then opening data files.

Conclusion

The two case-studies considered here are similar in that each involve articles which are linked to multiple data sets. For maximum convenience, it is optimal for researchers to be able to access this data without performing manual download and merging/extracting actions. There are also some differences between the two case-studies: in particular, the Parkinson's data spans multiple disciplines, whereas the Covid-19 data is more rigorously grounded in Flow Cytometry. As such, the operational requirements for the Covid-19 data, from a reader's point of view, are more clearly delineated: effective integration between the publication and its accompanying data sets is defined by launching Flow Cytometry software while a researcher is reading the publication, allowing the researcher to view the data via software similar to that used to generate/analyze the data while the reported research was being conducted. In the case of the Parkinson's data, in contrast, there is no single application which would seamlessly display the specrum of information considered in that article; as mentioned above, in such cases, SDRf would default to using QT Creator as a fallback for loading Supplemental Archives where no other software is available.

Regardless of whether one is using QT Creator or a domain-specific application, it is preferable that each SDRf archive be associated with one or more applications that researchers can use to view data (and extract information) from the archive. Moreover, these applications would ideally be linked to document viewers and also to publisher's portals, so that readers can automatically launch preferred applications and view Supplemental Archives while reading concomitant publications. In order to achieve this, scientific applications need to be augmented with plugins to parse SDRf data. To address this need, we are developing an inter-application messaging protocol so that disparate applications with SDRf plugins may interoperate. In particular, this protocol would entail PDF viewers being able to interoperate with scientific applications so that publications' data sets may be automatically downloaded and visualized via the preferred software.

Our prototype example for an application utilizing such plugins, as mentioned above, is software for Flow Cytometry. We are also working on a prototypic QT Creator plugin so that QT Creator (as the "default" SDRf software) can participate in SDRf networks using the same protocol. We will then expand the scope of this protocol via plugins for software in other domains, such as image-analysis, molecular visualization, radiology, 3D graphics, and so forth.