



A computational EXFOR database in JSON: Migration from MongoDB to CouchDB

Georg Schnabel

**Nuclear Data Section
International Atomic Energy Agency
Department for Nuclear Sciences and Applications**

Subgroup 49 – Reproducibility in Nuclear Data Evaluation – 13.5.2020

What's ahead?

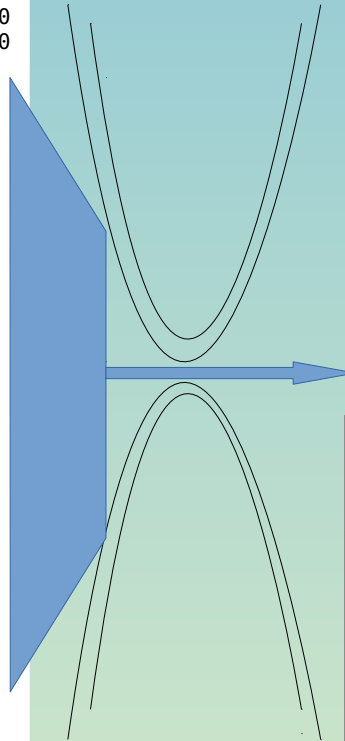
- EXFOR = JSON (recap from ND2019 in Beijing)
- Information extraction examples
- Document-oriented databases
- Examples of search and retrieval



EXFOR bottleneck

EXFOR format

```
ENTRY      10263   20011204   20020117   20050926   0000
SUBENT     10263001 20011204   20020117   20050926   0000
BIB        12      25
INSTITUTE  (1USAKTY)
REFERENCE  (J,NP/A,196,65,197211)
AUTHOR     (J.D.Brandenberger,A.Mittler,M.T.McEllistrem)
TITLE      The scattering of 8.0 and 9.0 MeV neutrons by 27Al,
            28Si, 31P and 32S
FACILITY   (VDG) Universty of Kentucky 6.0 MeV Van de Graaff
INC-SOURCE (D-D)
            (D-T)
METHOD     (TOF)
DETECTOR   (SCIN) Liquid organic scintillator.
MONITOR    (1-H-1(N,EL)1-H-1,,SIG) Normalized to n-p scattering.
ERR-ANALYS (DATA-ERR) Relative uncertainty given includes:
            . statistical uncertainties,
            . uncertainties in residual background,
            . statistical and stability uncertainties in monitor
              and electronics,
            . uncertainties in multiple scattering correction.
            . uncertainties in angular resolution correction.
            Normalization error 5.7%.
STATUS     Tabulated data received from J.D. Brandenberger,
            September 1972.
HISTORY    (19730202C)
            (19750214A) Silicon data added.
            (19821118A) Converted to REACTION formalism
            (20011204A) Converted to new date formats, lower case.
ENDBIB     25
NOCOMMON   0      0
ENDSUBENT  28      37
```



High-level languages
Python and R

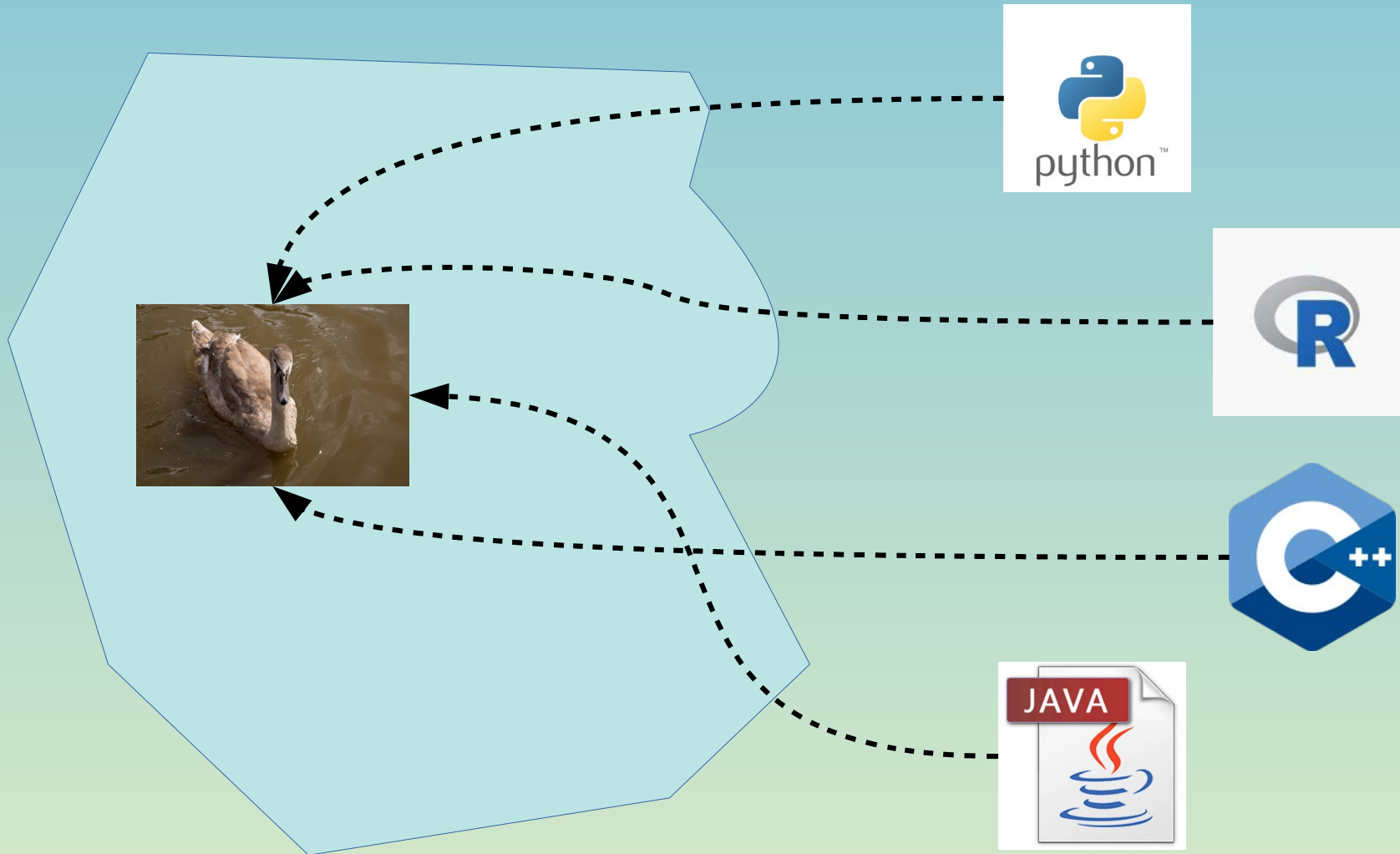
Many community
contributed packages
for machine learning



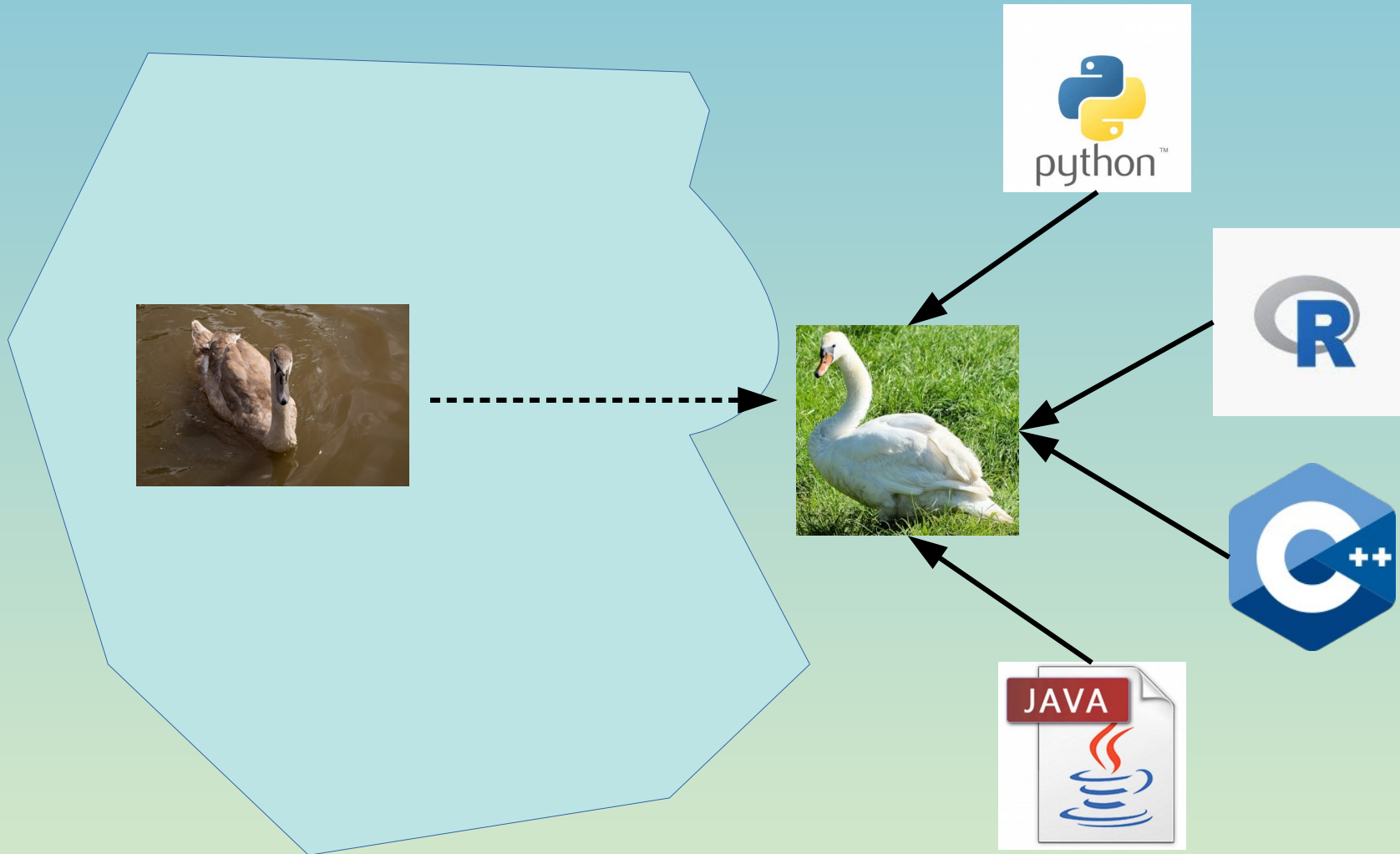
<https://www.business-science.io/business/2018/10/08/python-and-r.html>



Option #1: Teach languages EXFOR



Option #2: Bring EXFOR to languages



Hierarchical EXFOR

```

SUBENT      23171003   20170913   20180129   20180126   2265
BIB          11        26
REACTION    (26-FE-56(N,2N)26-FE-55,,SIG)
DECAY-DATA  (26-FE-55,2.73YR)
FACILITY    1(VDG,2ZZZGEL) Van de Graaff accelerator at IRMM.
            2(ACCEL,2AUSIRK) For AMS (accelerator mass spectrometry)
            at VERA lab. for radionuclides 10Be, 14C, 26Al, 55Fe,
            (CCW,2GERDRE)
METHOD      1(ACTIV) Activation technique combined with
            2(AMS) mass spectrometric technique
SAMPLE      Natural iron samples were irradiated at TU Dresden
            and IRMM.
INC-SOURCE1(D-T) T(d,n)He-4 .
INC-SPECT 1 Quasi-monoenergetic neutrons with energies between
            13.4 and 14.8 MeV;
            2 from 13 to 20 MeV.
FLAG        (1.) Experiment in 2007 yr,   TUD/VERA
            (2.) Experiment in 2010 yr,   TUD/VERA
            (3.) Experiment in 2010 yr,   IRMM/VERA
ERR-ANALYS (EN-ERR) finite neutron energy distribution and the
            absolute uncertainty in the neutron energy
            (namely it is EN-ERR+EN-RSL).
STATUS      (PRELM) Preliminary results (declared by A.Wallner,
            2017-08-30) under STATUS).
            (TABLE) Data received from the author (2017-08-30)
HISTORY      (20170913R) SD: Data received from the author were
            added. FLAG was added. BIB update according to
            comments from author.
ENDBIB      26
NOCOMMON    0          0
DATA        5          15
EN          EN-ERR    DATA    DATA-ERR    FLAG
MEV         MEV      MB        MB          NO-DIM
            13.35     0.15     300.      100.      3.
            13.49     0.04     322.4     16.1      1.
            ...
ENDDATA     17
ENDSUBENT   48
    
```

SUBENT

BIB

REACTION

FACILITY

. . .

DATA

UNITS

TABLE

Hierarchical format



EXFOR as JSON

```
{
  "ID": "23171003",
  "DATEMOD": "20170913",
  "BIB":{
    "REACTION": "(26-FE-56(N,2N)26-FE-55,,SIG)",
    "DECAY-DATA": "(26-FE-55,2.73YR)",
    "FACILITY": [
      "(VDG,2ZZZGEL) Van de Graaff accelerator ...",
      "(ACCEL,2AUSIRK) For AMS ... at VERA lab ..."
    ],
    "ERR-ANALYS": "(EN-ERR) finite neutron energy
      distribution and the absolute
      uncertainty in the neutron energy
      (namely it is EN-ERR+EN-RSL).",
    ...
  },
  ...
}
```

R Code for creation of JSON

```
library(exforParser)

filepath = "<PATH TO EXFOR FILE>"

exforText = readChar(filepath,
  file.info(filepath)$size)

entry = parseEntry(exforText)
convToJSON(entry$SUBENT[[3]])
```

<https://github.com/gschnabel/exforParser>



Convenient access

From R

```
...  
entry = parseEntry(exforText)  
entry$SUBENT[[3]]$BIB$REACTION  
  
entry$SUBENT[[3]]$DATA$UNIT  
  
entry$SUBENT[[3]]$DATA$TABLE
```

From Python

```
import json  
  
with open('exforEntryFile') as json_file:  
    entry = json.load(json_file)  
  
entry['SUBENT'][3]['BIB']['REACTION']  
  
...
```

From command line

```
jq ".SUBENT[3].BIB.REACTION" exforEntryFile.json  
  
jq ".SUBENT[3].DATA.TABLE.EN" exforEntryFile.json
```



Philosophy of EXFOR parser

- Keep it simple – keep it EXFOR, only introduce some commas, double-colons, and curly brackets to make it JSON
- Introduction of new fields, standardization of field values, etc. to get a better computational JSON format starting from raw EXFOR JSON
- Examples: standardization of units, merging of common blocks into data blocks, etc.

A computational EXFOR database, arXiv:1908.00209



Document-oriented databases

- Organized storage of JSON files
- Powerful search capabilities to locate and extract data



<https://www.mongodb.com>



<https://couchdb.apache.org/>



From MongoDB to CouchDB

- Originally creation of MongoDB EXFOR database*
- Due to controversial Server Side Public License migration to CouchDB under Apache 2.0 license
- As both database applications use JSON, migration was straight-forward

*) <https://github.com/gschnabel/compEXFOR-docker>



Glimpse into CouchDB API

- API accessible via http-request (“visiting a web page”)
- Every search specification is in JSON
- Search JSON is an incomplete skeleton of the EXFOR subentry
- Returned results are in JSON

```
{  
  "selector": {  
    "BIB": {  
      "AUTHOR": {  
        "$regex": "FERGUSON"  
      }  
    }  
  }  
}
```



Example of retrieval from CouchDB

Retrieve EXFOR subentry (11464003) on command line

```
curl -X GET http://admin:password@localhost:5984/exfor/11464003 > result.json
```

Extract specific fields on command line

```
jq -r '.BIB.AUTHOR' result.json
```

```
jq -r '.DATA.TABLE.EN[]' result.json
```

```
jq -r '.DATA.TABLE.DATA[]' result.json
```



Example of search using CouchDB

**Finding EXFOR subentries with participation of FERGUSON
(limited to first four results)**

```
curl -s -X POST --header "Content-Type: application/json" \  
  -d '{"selector": {"BIB": {"AUTHOR": { "$regex": "FERGUSON"}}}, "limit": 4 }' \  
  http://admin:password@localhost:5984/exfor/_find > result.json
```

<https://github.com/iaea-nds/exfor-couchdb-docker>



Conclusions

- EXFOR to JSON conversion makes access to all information in EXFOR easier
- Storing EXFOR information in a document-oriented database application enables flexible searching (e.g., also searching in free text) and retrieval of information
- Ease of access facilitates the creation of derived databases for statistical analysis / machine learning and the creation of scripts for reproducible nuclear data evaluation



Acknowledgments / Alternatives

- **x4toc4** by D.E. Cullen
- **ENDVER** by A. Trkov
- **x4i** interface by D. Brown (and fork **x4i3** by A. Fedynitch)
- **EXFOR web system** (with JSON retrieval) by V.V. Zerkin and B. Pritchychenko
- **EXFORTables** by A. Koning
- **EXFOR parser** for Wolfram language by E. Teixeira



Backup slides



Specific examples (command line)

Getting an EXFOR subentry in JSON from command line using, e.g., data from subentry ID 11464003:

```
curl -X GET http://admin:password@localhost:5984/exfor/11464003 | jq .
```

Extracting, e.g., the AUTHOR field from this subentry

```
curl -s -X GET http://admin:password@localhost:5984/exfor/11464003 | jq -r '.BIB.AUTHOR'
```

Get available columns in the data table:

```
curl -s -X GET http://admin:password@localhost:5984/exfor/11464003 | jq -r '.DATA.DESCR[]'
```

and their units (everything is normalized to MeV and millibarn in the database):

```
curl -s -X GET http://admin:password@localhost:5984/exfor/11464003 | jq -r '.DATA.UNIT[]'
```

Get all the numbers in the DATA column:

```
curl -s -X GET http://admin:password@localhost:5984/exfor/11464003 | jq -r  
'DATA.TABLE.DATA[]'
```



Specific examples (command line)

Finding EXFOR subentries with participation of FERGUSON (limited to first four results):

```
curl -s -X POST --header "Content-Type: application/json" -d '{"selector": {"BIB": {"AUTHOR": {"$regex": "FERGUSON"}}}, "limit": 4 }' http://admin:password@localhost:5984/exfor/_find | jq .
```

Only extract only particular fields, e.g., AUTHOR, REACTION and DATA column:

```
curl -s -X POST --header "Content-Type: application/json" -d '{"selector": {"BIB": {"AUTHOR": {"$regex": "FERGUSON"}}}, "limit": 4, "fields": ["BIB.AUTHOR", "BIB.REACTION", "DATA.TABLE.DATA"]}' http://admin:password@localhost:5984/exfor/_find | jq .
```

Simple computational format: output the two columns EN and DATA separated by tabulator (assuming it is the same quantity in the subentries, which it is not the case in this example):

```
curl -s -X POST --header "Content-Type: application/json" -d '{"selector": {"BIB": {"AUTHOR": {"$regex": "FERGUSON"}}}, "DATA": {"TABLE": {"EN": {"$exists": true}, "DATA": {"$exists": true}}} }, "limit": 4}' http://admin:password@localhost:5984/exfor/_find | jq -r '[[([.docs[].DATA.TABLE.EN] | reduce .[] as $x ([], . + $x)), ([.docs[].DATA.TABLE.DATA] | reduce .[] as $y ([], . + $y))] | transpose | .[] | @tsv'
```

<https://github.com/iaea-nds/exfor-couchdb-docker>

