

Accessing training data for machine-learning: materials data APIs

Cormac Toher

Department of Materials Science and Engineering

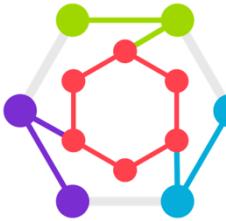
Department of Chemistry and Biochemistry

The University of Texas at Dallas

(cormac.toher@utdallas.edu)



AFLOW
Automatic-FLOW for Materials Discovery



OPTIMADE
Open Databases Integration
for Materials Design



NOVEL MATERIALS DISCOVERY

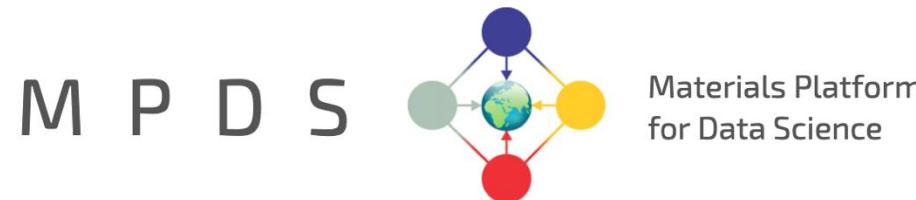


OQMD



Accessing online data

- Machine-learning benefits from large quantities of data for training and testing
- Materials data is now available online through websites and programmatically accessible APIs
- Available data includes both experimental (relatively scarce and often requiring license fees) and computational data (relatively plentiful and freely available)
- Experimental materials phase data: Materials Platform for Data Science (commercial):
<https://developer.mpds.io/>



- Experimental materials properties data (e.g. superconductivity): Materials Data Repository:
<https://mdr.nims.go.jp/?locale=en>

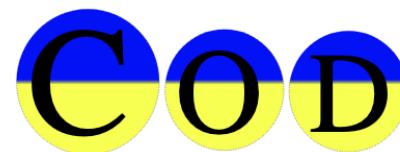


Accessing online data

- Crystallography data:
 - Inorganic Crystal Structure Database (requires purchasing subscription):
<https://icsd.nist.gov/>

NIST Inorganic Crystal Structure Database (ICSD)

- Crystallography Open Database (open-access):
<http://www.crystallography.net/cod/>

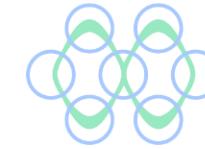


Crystallography Open Database

Accessing online data

- Computational materials data:

- Materials Project: <https://materialsproject.org/>



- AFLOW: <http://aflow.org/>



- OQMD: <https://oqmd.org/>

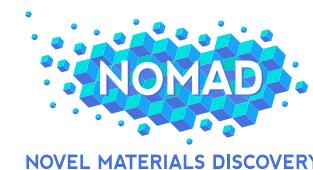


- Materials Cloud: <https://www.materialscloud.org/>



- Materials Project Descriptor Database: <http://mpdd.phaseslab.com/>

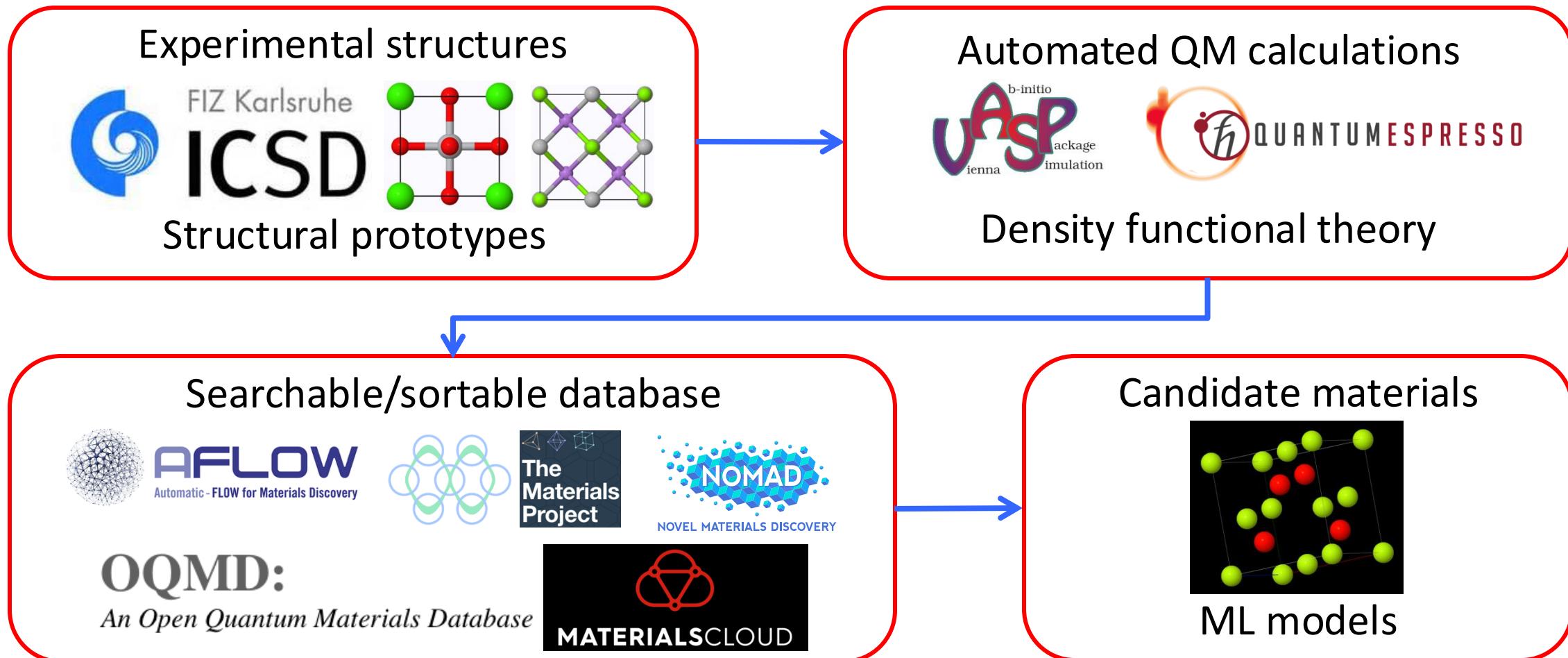
- NOMAD: <https://nomad-lab.eu/>



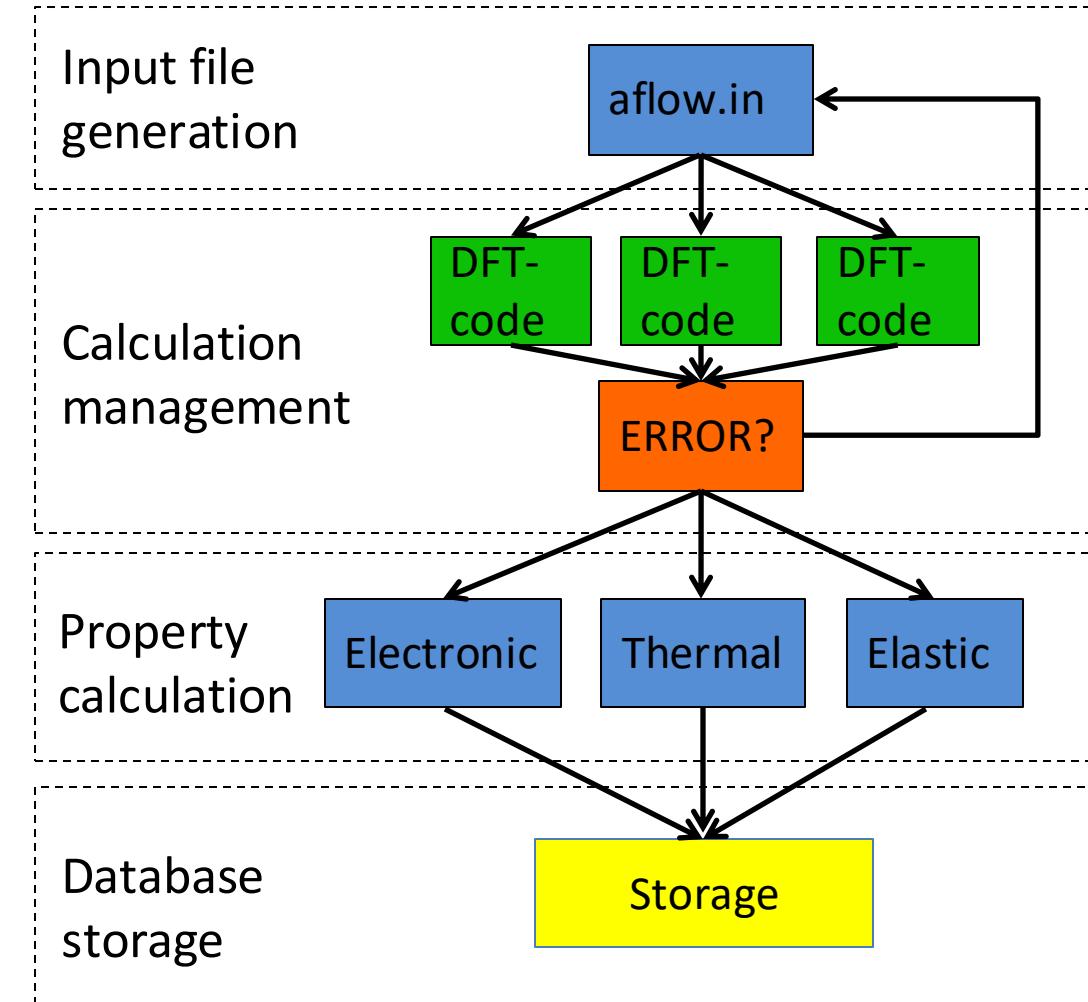
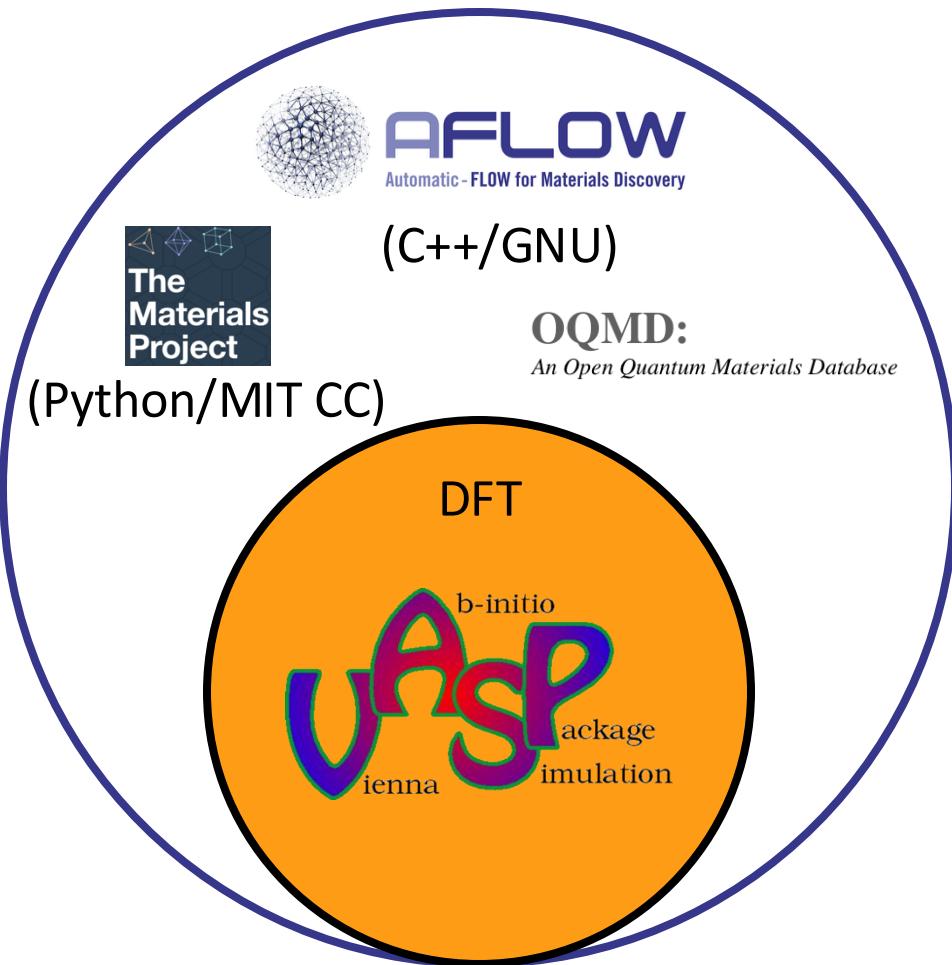
- Common API for materials data: Optimade: <https://www.optimade.org/>



Automated Computational Materials Discovery



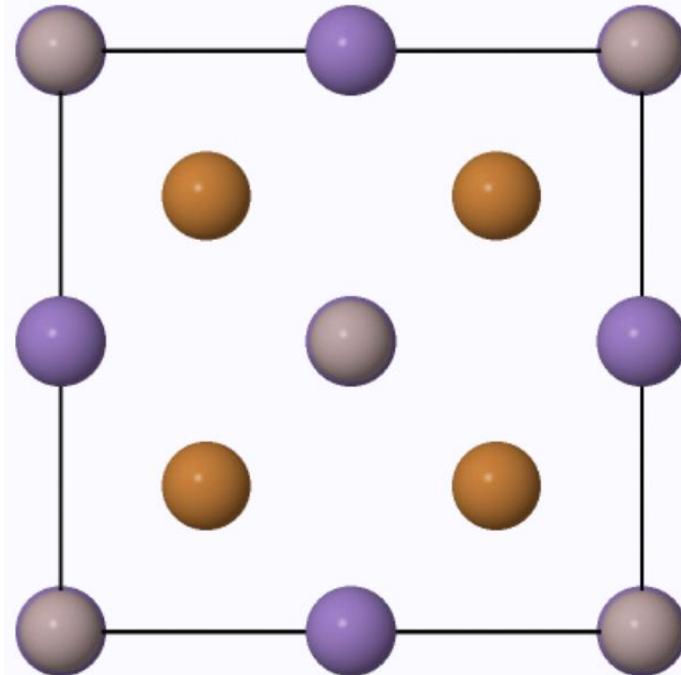
Automated Computational Materials Discovery



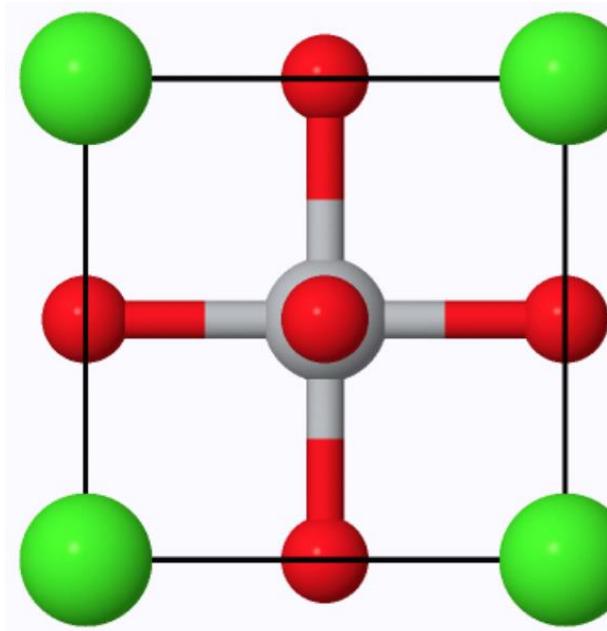
Curtarolo et al., Comput. Mater. Sci. **58**, 218 (2012); Toher et al., “The AFLOW Fleet for Materials Discovery” (2018);
Oses et al., Comput. Mater. Sci. **217**, 111889 (2023)

Crystallographic Prototypes

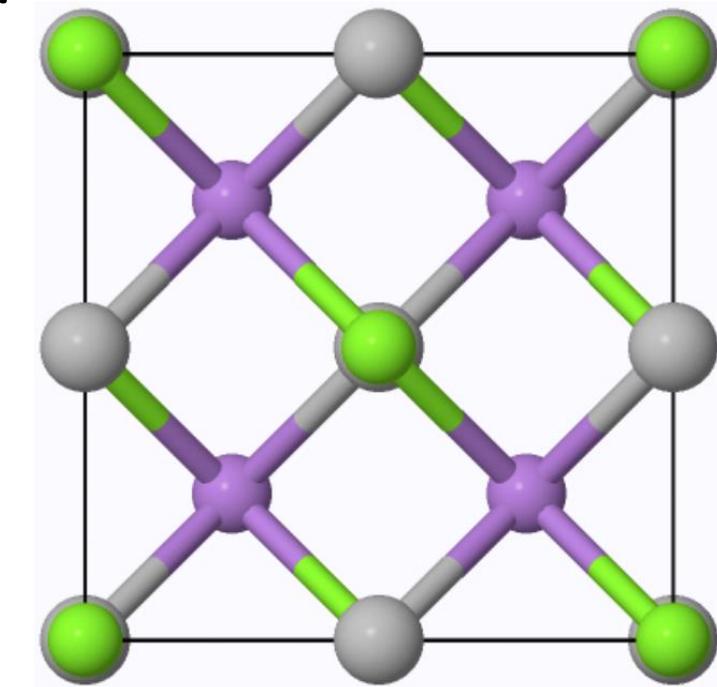
- Search for new materials using structural prototypes:



Heusler structure:
AB₂C_cF16_225_a_c_b



Perovskite structure:
ABC₃_cP5_221_a_b_c



Half-Heusler structure:
ABC_cF12_216_b_c_a

- Decorate prototypes with different elements to generate new hypothetical materials

Mehl et al., Comput. Mater. Sci. **136**, S1-S828 (2017); Hicks et al., Comput. Mater. Sci. **161**, S1-S1011 (2019);
Hicks et al., npj Comput. Mater. **7**, 30 (2021).

AFLOW Web Portal

<https://aflow.org>



HOME | CONSORTIUM | SEMINARS | SCHOOLS | FORUM | SRC

> 700 million properties

AFLOW Schools and Seminars

> 3.5 million entries

breakdown of properties

AFLOW Seminars

AFLOW Schools

Welcome to AFLOW, a globally available database of **3,530,330** material compounds with over **734,308,640** calculated properties, and growing.

3,479,057
form. enthalpies

366,988
band structures

172,488
Bader charges

5,650
elastic properties

5,664
thermal properties

1,738
binary systems

30,289
ternary systems

150,659
quaternary systems

Quick Search

Enter an ICSD Number, [Aflowlib Unique Identifier](#), or advanced search string (e.g. Fe & Si).

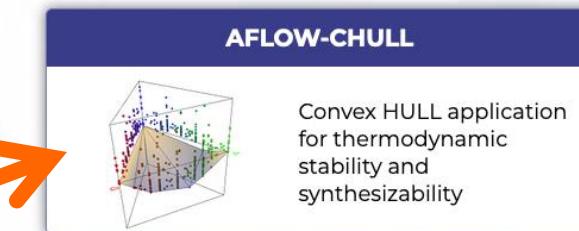
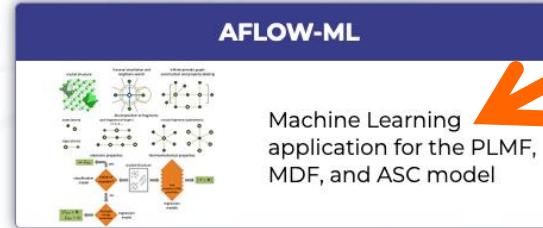
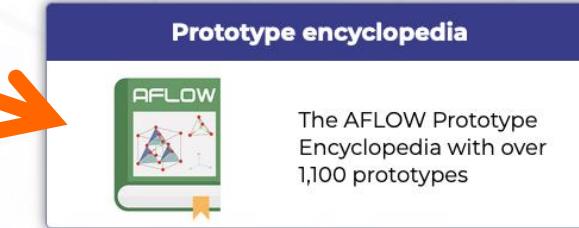
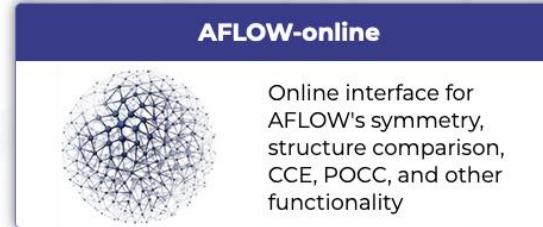
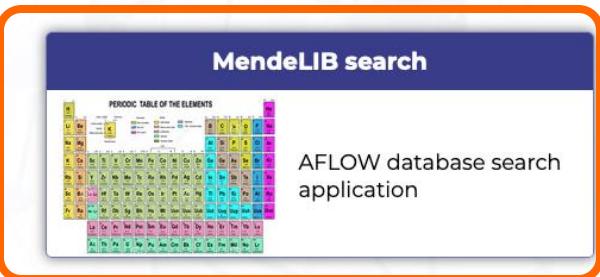
ICSD#, AUID#, search string...

Search

AFLOW Web Portal

Apps and Docs

Crystal
prototype
library



Thermodynamic
stability

ML
application

AFLOW Search Page

The screenshot shows the AFLOW search interface. At the top, there is a navigation bar with links to HOME, CONSORTIUM, PUBLICATIONS, FORUM, SRC, and SEARCH. Below the navigation bar is a search bar with options for "ICSD only" and "All AFLOW". To the right of the search bar is a green "Search (60379 entries)" button and a "Display" toggle switch. The main area features a periodic table with various filters and logic operations.

Element group selection: A callout box labeled "Element group selection" points to the left side of the periodic table, specifically highlighting the first column (Metals) and the second column (Alkali Metals).

Logic operations: A callout box labeled "Logic operations" points to the logic operators section above the periodic table, which includes "and", "not", "or", "xor", and parentheses.

Number of species: A callout box labeled "Number of species" points to the "# of species" filter section on the right side of the periodic table.

Element selection: A callout box labeled "Element selection" points to the center of the periodic table, specifically highlighting the element Ar (Argon).

AFLOW Search Page

The screenshot shows the AFLOW search interface. At the top, there's a navigation bar with links to HOME, CONSORTIUM, PUBLICATIONS, FORUM, SRC, and SEARCH. Below the navigation bar is a search bar with the placeholder "Search Aflow" and two buttons: "ICSD only" and "All AFLow". To the right of the search bar is a green button labeled "Search (60379 entries)" and a "Display" toggle switch, which is highlighted with a red arrow. The main area features a large periodic table grid with elements color-coded by group. Above the periodic table are three sections: "Property Filters" on the left, "Periodic Table" in the center, and another "Property Filters" section on the right. The "Property Filters" sections contain various search operators like "and", "not", "or", "xor", and parentheses, along with a dropdown for "# of species". At the bottom of the page, there are logos for Duke University, University at Buffalo, Missouri S&T, NC State University, Penn State, NSF, and ONR.

Slide to Property Filters

Bring property filters onto search page

AFLOW Search Page

The screenshot shows the AFLOW search interface. At the top, there's a navigation bar with links to HOME, CONSORTIUM, PUBLICATIONS, FORUM, SRC, and SEARCH. Below the navigation is a search bar with dropdowns for "Search Aflow" (set to "ICSD only") and "Display" (set to "All AFLOW"). A green "Search" button is visible, along with a note that "(60379 entries)" have been found. To the right of the search bar is a "Display" toggle switch.

Below the search bar is a "Periodic Table" section. It features a standard periodic table grid with element symbols and numbers. Above the table are several filter categories: Metals, Alkali Metals, Alkali Earths, Transition Metals, Lanthanides, Other Metals, Non-Metals, Boron Group, Carbon Group, Pnictogens, Chalcogens, and Halogens. To the right of the table are additional filter options: "and" or "not", "# of species" (with dropdowns for = and >), "or" or "xor", and parentheses for complex queries.

At the bottom of the page is a "Property Filters" section. It includes a "Search Properties" input field and a "Clear all" button. The filters are organized into sections: Electronic Properties, Magnetic Properties, Mechanical Properties, Thermodynamic Properties, Relaxed Structure, Chemistry, and Calculation Details. Under Electronic Properties, there are filters for "band gap type" (describing metals, semi-metals, or insulators with direct or indirect band gaps) and "electronic band gap" (describing electronic band gaps). Under Magnetic Properties, there is a filter for "Pearson symbol". Each filter has checkboxes for "Display column" and "Restrict value" or "equal to" options.

Property Filters

AFLOW Search Page

The screenshot shows the AFLOW search interface. At the top, the search bar contains "Na, Cl". Below it, under "ICSD only", there is a "Search" button (60379 entries) and a "Display" toggle. A periodic table is centered, with the search terms "Na, Cl" highlighted. To the right of the table, a search filter panel is open, showing the following settings:

- Space group:** Display column
 Restrict value from 1 to 230
- Pearson symbol:** Display column
 Restrict value equal to
- Electronic band gap:** Display column
 Restrict value from 0 to 17.6792 eV

Annotations with orange arrows and boxes explain the search process:

- Selected Elements:** Points to the search bar with "Na, Cl".
- Number of species=2**: Points to the "# of species" field showing "2".
- Added filter for Egap: electronic band gap**: Points to the "electronic band gap" section of the filter panel.
- Select “Add” to add filter**: Points to the "Add" button in the filter panel.
- Check box to restrict search results to specific value range**: Points to the "Restrict value" fields in the filter panel.

AFLOW Search Results

Na , Cl

ICSD only All AFLW x

Search (60389 entries) **Display**

Reset Search

Aflux summons: [https://aflux.org/API/aflux/?species\(Na,Cl\),\\$catalog\(ICSD\),\\$nspecies\(2\),Egap\(*\),\\$paging\(1,1000\)](https://aflow.org/API/aflux/?species(Na,Cl),$catalog(ICSD),$nspecies(2),Egap(*),$paging(1,1000))

Structural Information **Requested Properties**

Results per page: 20 Select page: 1 Found 27 entries

ENTRY	space group relaxed	Pearson symbol relaxed	electronic band gap (eV)	DATA
CINa [1354bbef4edd80b3]	$Fm\bar{3}m$ (#225)	cF8	5.06	[API, Out, JSON]
CINa [18ebb85b07a92f89]	$Fm\bar{3}m$ (#225)	cF8	5.06	[API, Out, JSON]
CINa [182f848dd10cc403]	$Fm\bar{3}m$ (#225)	cF8	5.06	[API, Out, JSON]
CINa [1cd71114972d46dd]	$Fm\bar{3}m$ (#225)	cF8	5.04	[API, Out, JSON]
CINa [2f4b5e32510830a0]	$Fm\bar{3}m$ (#225)	cF8	5.05	[API, Out, JSON]
CINa [39ab5e62afdb5ac0]	$Fm\bar{3}m$ (#225)	cF8	5.05	[API, Out, JSON]
CINa [4f19021768a3118a]	$Fm\bar{3}m$ (#225)	cF8	5.05	[API, Out, JSON]

↑

AFLOW Search Page: Exercises

1. Use the AFLOW Advanced Search Page to find the electronic band gap of SiC with the zincblende structure (space group 216)
2. Use the AFLOW Advanced Search Page to find the electronic band gap of GaN with the wurtzite structure (space group 186)

Materials Project Web Portal



The Materials Project

Apps About ▾ Community ▾ ML API Login



The Materials Project

Harnessing the power of supercomputing and state-of-the-art methods, the Materials Project provides open web-based access to computed information on known and predicted materials as well as powerful analysis tools to inspire and design novel materials.

Login or Register



MP login

- Materials Project database requires registration/login
- Can login using Google/Facebook/Github/Amazon/Microsoft account

Materials Project Web Portal

The Materials Project

Apps About Community ML API

The Materials Project

Harnessing the power of supercomputing and state-of-the-art methods, the Materials Project provides open web-based access to computed information on known and predicted materials as well as powerful analysis tools to inspire and design novel materials.

[Start Exploring Materials](#) [Search Materials](#) [Browse Apps](#)

MP Materials Explorer

- Select Materials Explorer application to search for materials

Materials Explorer Application

The Materials Project

Apps ▾ About ▾ Community ▾ ML API ⚙

Home / Apps / Materials Explorer

Materials Explorer
App by Materials Project

Search for materials information by chemistry, composition, or properties.

Materials e.g. Li-Fe or Li₂Fe or Li₃Fe or mp-19017

?

Search

Only Elements At Least Elements Formula

* Select elements to search for materials with **only** these elements

H	Li	Be		B	C	N	O	F	He										
Na	Mg		Al	Si	P	S	Cl	Ne											
K	Ca	Sc	Ti	V	Cr	Mn	Fe	Co	Ni	Cu	Zn	Ga	Ge	A	Sb	Te	I	Br	Kr
Rb	Sr	Y	Zr	Nb	Mo	Tc	Ru	Rh	Pd	Ag	Cd	In	Sn	Sb	Te	I	Xe		
Cs	Ba	La-Lu	Hf	Ta	W	Re	Os	Ir	Pt	Au	Hg	Tl	Pb	Bi	Po	At	Rn		
Fr	Ra	Ac-Lr	Rf	Db	Sg	Bh	Hs	Mt	Ds	Rg	Cn	Nh	Fl	Mc	Lv	Ts	Og		
La	Ce	Pr	Nd	Pm	Sm	Eu	Gd	Tb	Dy	Ho	Er	Tm	Yb	Lu					
Ac	Th	Pa	U	Np	Pu	Am	Cm	Bk	Cf	Es	Fm	Md	No	Lr					

Other explorer applications

Enter elements or chemical formula

Filter by only selected elements, at least elements, or chemical formula

Element selection

Materials Explorer: Search Results

Filters Reset

- ▶ Composition 1 active
- ▶ Thermodynamics
- ▶ Structural Properties
- ▶ Symmetry
- ▶ Calculated Properties
- ▶ Electronic Structure
- ▶ Magnetism
- ▶ Elasticity
- ▶ Surfaces

Property Filters

13 materials match your search
Showing 1-13

Columns ▾

×

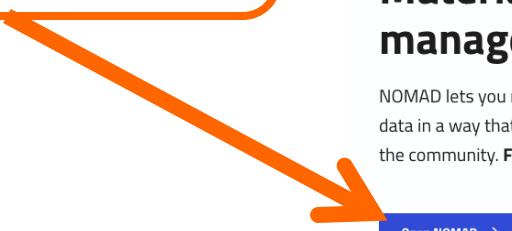
Chemical System: Na-Cl

Material ID	Formula	Crystal System	Space Group Symbol	Sites	Energy Above Hull (eV/atom)	Band Gap (eV)
★ mp-22862	NaCl	Cubic	Fm3m	2	0	5.00
mp-1189265	NaCl ₃	Orthorhombic	Pnma	16	0	2.71
mp-1080771	NaCl ₇	Cubic	Pm3	8	0	0.43
mp-1120767	NaCl	Hexagonal	P6 ₃ /mmc	12	0.06	4.38
mp-990084	Na ₂ Cl	Trigonal	R3m	3	0.10	0
mp-1064484	Na ₃ Cl	Tetragonal	P4/mmm	4	0.14	0
mp-1018810	Na ₂ Cl	Tetragonal	P4/mmm		0.14	0
★ mp-22851	NaCl	Cubic	Pm3m	2	0.15	3.97
mp-1069676	Na ₃ Cl ₂	Orthorhombic	Cmmm			

Structural Information

NOMAD: Novel Materials Discovery

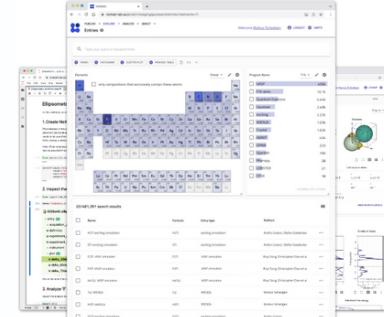
Search application



NOMAD
Materials science data managed and shared

NOMAD lets you manage and share your materials science data in a way that makes it truly useful to you, your group, and the community. **Free and open source.**

[Open NOMAD →](#)



NOMAD Discord

Join our brand new NOMAD Discord, where you can connect with fellow researchers, share ideas, and get answers.

[Open Discord](#)



USED BY THOUSANDS OF MATERIALS SCIENTISTS

UPLOADED ENTRIES
19,097,756

REPRESENTED MATERIALS
4,335,633

UPLOADED FILES
111.9 TB

NOMAD: Novel Materials Discovery

The screenshot shows the NOMAD interface with several orange annotations:

- An arrow points from the text "Element selection" to the periodic table area.
- A callout box labeled "Search by elements" points to the search bar above the periodic table.
- A callout box labeled "Return only entries with only these elements" points to the checkbox in the search bar that filters for compositions containing only the selected atoms.

FILTERS

- Material
- Elements / Formula only compositions that exclusively contain these atoms
- Structure / Symmetry
- Method
- Precision
- DFT
- TB
- GW
- BSE
- DMFT
- EELS
- Workflow
- Molecular dynamics
- Geometry Optimization
- Properties
- Electronic
- Vibrational
- Mechanical

ELEMENTS / FORMULA

Elements

Chemical formula hill

Chemical formula IUPAC

Chemical formula reduced

Chemical formula anonymous

Upload time ↓

Authors

Upload time	Authors
1/2/2025, 4:55:20 AM	Cecilia Vona
12/19/2024, 6:12:36 AM	Miguel Marques

NOMAD: Novel Materials Discovery

The screenshot shows the NOMAD interface. On the left is a sidebar with 'FILTERS' and various search categories like Material, Elements / Formula, Structure / Symmetry, Method, Precision, DFT, TB, GW, BSE, DMFT, EELS, Workflow, Molecular dynamics, Geometry Optimization, Properties, Electronic, Vibrational, Mechanical, Use Cases, and Solar Cells. A red arrow points from a box labeled 'Search query' to the search bar at the top, which contains the text 'Na AND Cl'. Another red arrow points from a box labeled 'Search results' to the list of search results below.

Search query

Search results

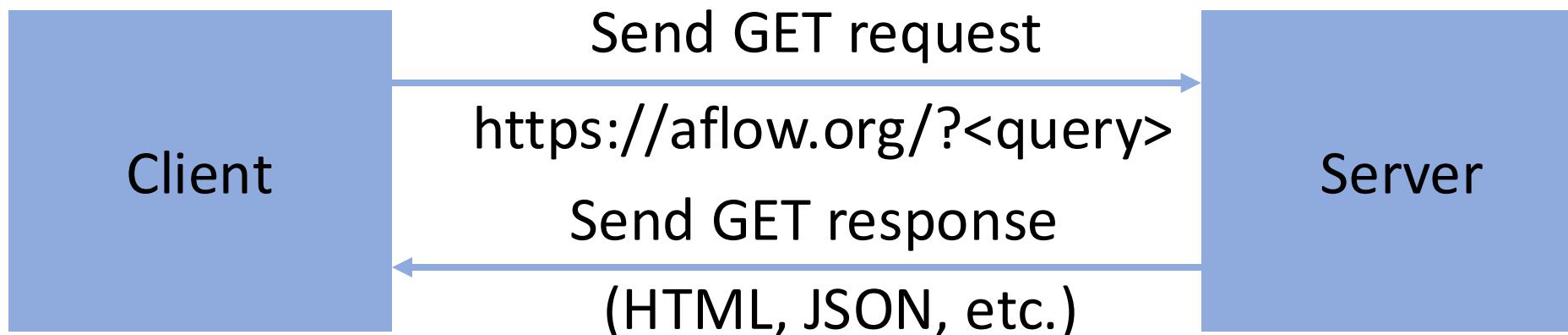
Name	Formula	Entry type	Upload time	Authors
NaCl exciting DFT SinglePoint simulation	CINa	exciting DFT SinglePoint	1/2/2025, 4:55:20 AM	Cecilia Vona
NaCl exciting DFT SinglePoint simulation	CINa	exciting DFT SinglePoint	1/2/2025, 4:55:20 AM	Cecilia Vona
NaCl exciting DFT SinglePoint simulation	CINa	exciting DFT SinglePoint	1/2/2025, 4:55:20 AM	Cecilia Vona
NaCl exciting DFT SinglePoint simulation	CINa	exciting DFT SinglePoint	1/2/2025, 4:55:20 AM	Cecilia Vona
NaCl exciting DFT SinglePoint simulation	CINa	exciting DFT SinglePoint	1/2/2025, 4:55:20 AM	Cecilia Vona
NaCl exciting DFT SinglePoint simulation	CINa	exciting DFT SinglePoint	1/2/2025, 4:55:20 AM	Cecilia Vona
NaCl3 VASP GeometryOptimization simulation	Cl6Na2	VASP GeometryOptimization	12/19/2024, 6:12:36 AM	Miguel Marques
NaCl3 VASP GeometryOptimization simulation	Cl6Na2	VASP GeometryOptimization	12/19/2024, 6:12:36 AM	Miguel Marques
NaCl3 VASP DFT SinglePoint simulation	Cl6Na2	VASP DFT SinglePoint	12/19/2024, 6:12:36 AM	Miguel Marques
	Cl6Na2	VASP GeometryOptimization	12/19/2024, 6:12:36 AM	Miguel Marques

Programmatic access to online data

- Specific materials or molecules can be searched for directly using web portals
- However, machine-learning models often require large data sets for training on
- Therefore, it is useful to be able to search for and download data programmatically
- This is performed through an Application Programming Interface (API)
- Examples: Materials Project API (for Materials Project API); AFLUX (for AFLOW database: aflow.org); AiiDA (for Materials Cloud); Optimade API (common API to enable access to multiple databases)

Accessing online data: REST-API

- REST-API: Representational State Transfer Application Programming Interface
- In response to a client request, the API transfers a representation of the current state (e.g. current version of a section of a database) to the client
- Commonly used methods: GET (retrieve data from server), POST (post data to server)



Accessing online data: REST-API

- GET request can be submitted in the form of URI (Universal Resource Identifier) through a browser, using urllib module of Python, or using UNIX utilities such as wget or cURL
- URI has the following structure:
$$<\text{scheme}>://<\text{authority}>/<\text{path}>?<\text{query}>\#<\text{fragment}>$$

server query
- Example: AFLOW AFLUX Search-API: <https://aflow.org/API/aflux/?<query>>

server query
- Query part can contain materials keywords with associated value ranges to filter returned entries

Accessing online data: REST-API

- Use Python to submit GET request:

```
import modules for JSON, sys, os  
import json, sys, os  
from urllib.request import urlopen  
  
Import modules for urlopen  
Server path  
SERVER="https://" + <server> + "?"  
Search query  
QUERY=<query>  
URI  
EXAMPLE_URI = SERVER + QUERY  
Submit URI with query  
  
response=json.loads(urlopen(EXAMPLE_URI).read().decode("utf-8"))  
print(response)  
Read/decode JSON response  
Convert to dictionary-like object  
Print dictionary-like object
```

AFLUX: AFLOW Search API

`https://aflow.org/API/aflux/?<matchbook>, <directives>`

Search API Server

Query

Matchbook

- Materials keywords with arguments
- Keyword list is available from the help directive:

Directives

- Formatting instructions with arguments
- Available directives:
 - format: json (default) or html
 - paging: Controls number of entries and page displayed

[https://aflow.org/API/aflux/?help\(properties\)](https://aflow.org/API/aflux/?help(properties))

<https://aflow.org/API/aflux/?schema>

F. Rose et al., Comput. Mater. Sci. **137**, 362 (2017).

AFLUX: AFLOW Search API

- AFLUX supports use of several logical operators
- Operator scope can be inter-property and/or intra-property

Logical operator	AFLUX syntax	Operator scope
<block> (precedence)	"(" and ")"	Intra- and inter-property
<AND>	","	Intra- and inter-property
<OR>	Intra- and inter-property	
<NOT>	"!"	Intra-property
<loose>	"*"	Intra-property
<string>	"'"	Inter-property
<mute>	"\$"	Intra-property

AFLUX: AFLOW Search API

- Common keywords:

species: chemical species (usually element symbols)

nspecies: number of different species in compound

Egap: electronic band gap

spacegroup_relax: Space group of relaxed structure

aflow_prototype_label_relax: AFLOW prototype label (structure type) of relaxed structure

- Example: NaCl or KCl with an electronic band gap of at least 5 eV

[https://aflow.org/API/aflux/?species\(\(Na:K\),Cl\),nspecies\(2\),Egap\(5*\)](https://aflow.org/API/aflux/?species((Na:K),Cl),nspecies(2),Egap(5*))

server

(Na OR K) AND Cl

Number of
species == 2

Band gap
> 5eV

AFLUX: AFLOW Search API

- Data is returned in JSON format: JavaScript Object Notation
- JSON files can be read in and converted to Python dictionaries:

```
import json  
response_dict = json.load(response)
```

- Results are sorted in ascending order of first keyword in query
- Example: to sort in order of band gap:

[https://aflow.org/API/aflux/?Egap\(5*\),species\(\(Na:K\),Cl\),nSpecies\(2\)](https://aflow.org/API/aflux/?Egap(5*),species((Na:K),Cl),nSpecies(2))

AFLUX: AFLOW Search API

- Example: NaCl or KCl with an electronic band gap of at least 5eV

```
https://aflow.org/API/aflux/?species((Na:K),Cl),nspecies(2),  
Egap(5*)
```

```
import json, sys, os  
from urllib.request import urlopen
```

Import modules for JSON, sys, os
Import modules to open URLs

```
SERVER="https://aflow.org"  
API="/API/aflux??"
```

AFLOW server
AFLUX API

```
MATCHBOOK="species((Na:K),Cl),nspecies(2),Egap(5*)"
```

AFLUX matchbook

```
DIRECTIVES="$paging(0)"
```

AFLUX directives

```
SUMMONS=MATCHBOOK+","+DIRECTIVES
```

AFLUX query

```
response=json.loads(urlopen(SERVER+API+SUMMONS).read().decode("utf-8"))  
print(response)
```

Submit query, retrieve json, convert to Python dictionary
Print Python dictionary

AFLUX: AFLOW Search API

- Example: NaCl or KCl with an electronic band gap of at least 5eV

```
https://aflow.org/API/aflux/?species((Na:K),Cl),nspecies(2),Egap(5*)
```

Programmatically searching for data using AFLUX API: search for NaCl or KCl with an electronic band gap greater than 5eV

```
✓ 5s  ➡ https://aflow.org/API/aflux/?species((Na:K),Cl),nspecies(2),Egap(5*)
```

```
import json, sys, os
from urllib.request import urlopen

SERVER="https://aflow.org"
API="/API/aflux/?"
MATCHBOOK="species((Na:K),Cl),nspecies(2),Egap(5*)"
DIRECTIVES="$paging(0)"
SUMMONS=MATCHBOOK+","+DIRECTIVES

print(SERVER+API+SUMMONS)

response=json.loads(urlopen(SERVER+API+SUMMONS).read().decode("utf-8"))
print(response)
```

```
→ https://aflow.org/API/aflux/?species((Na:K),Cl),nspecies(2),Egap(5*),$paging(0)
[{'compound': 'Cl1K1', 'auid': 'aflow:04b2e3ac1bd9b49b', 'aurl': 'aflowlib.duke.edu:AFL0WDATA/ICSD_WEB/FCC/Cl1K1_ICSD_240523'},
```

AFLUX: AFLOW Search API

- Example: NaCl or KCl with an electronic band gap of at least 5eV
- Parse output: print chemical formula and band gap

```
response=json.loads(urlopen(SERVER+API+SUMMONS).read().decode("utf-8"))
for datum in response: ← Loop over entries in dictionary
    bandgap=[float(datum['Egap'])] ← Extract band gap
    compound=[str(x) for x in datum['compound'].split(",")]
    print ("{}, {}".format(compound, bandgap)) ← Extract compound
    ← Print data
```

AFLUX: AFLOW Search API

- Example: NaCl or KCl with an electronic band gap of at least 5eV
- Parse output: print chemical formula and band gap

Parse JSON response: print chemical formula and electronic band gap

```
✓ 4s
  import json, sys, os
  from urllib.request import urlopen

  SERVER="https://aflow.org"
  API="/API/aflux/?"
  MATCHBOOK="species((Na:K),Cl),nspecies(2),Egap(5*)"
  DIRECTIVES="$paging(0)"
  SUMMONS=MATCHBOOK+","+DIRECTIVES

  response=json.loads(urlopen(SERVER+API+SUMMONS).read().decode("utf-8"))
  for datum in response:
      bandgap=[float(datum['Egap'])]
      compound=[str(x) for x in datum['compound'].split(",")]
      print ("{} , {}").format(compound, bandgap))

  ['Cl1K1'], [5.0487]
  ['Cl1K1'], [5.0534]
  ['Cl1K1'], [5.0547]
  ['Cl1K1'], [5.0529]
```

AFLUX: AFLOW Search API

- Directives: paging – control page number and number of entries per page: useful to limit amount of data returned at once to avoid overloading browser/computer
- “Page” is set of data returned at a time; “**paging(0)**” returns all data
- Paging takes two arguments: first is page number “**n**”, second is number of entries per page “**m**”: **paging(n,m)**
- Example: return first page with 10 entries per page: **paging(1,10)**
- Negative page number reverses sort order: **paging(-1,10)**

AFLUX: AFLOW Search API

- Example: NaCl or KCl with an electronic band gap of at least 5eV: first page of 10 entries

Paging: print first page of 10 entries per page

```
▶ import json, sys, os
  from urllib.request import urlopen

  SERVER="https://aflow.org"
  API="/API/aflux/?"
  MATCHBOOK="species((Na:K),Cl),nspecies(2),Egap(5*)"
  DIRECTIVES="$paging(1,10)"
  SUMMONS=MATCHBOOK+","+DIRECTIVES

  response=json.loads(urlopen(SERVER+API+SUMMONS).read().decode("utf-8"))
  for datum in response:
    bandgap=[float(datum['Egap'])]
    compound=[str(x) for x in datum['compound'].split(",")]
    print ("{}, {}".format(compound, bandgap))

⇒ ['Cl1K1'], [5.0427]
  ['Cl1K1'], [5.0482]
  ['Cl1K1'], [5.0422]
```

AFLUX: AFLOW Search API

- Example: search for materials containing C, Si, Ge, Sn, but not Pb:

```
<API>/?species( (C:Si:Ge:Sn) , !Pb) , $paging(1,64)
```

C or Si or Ge or Sn

NOT Pb

First page of 64 entries

Search for materials containing C, Si, Ge or Sn but not Pb

```
[2] import json, sys, os
     from urllib.request import urlopen

     SERVER="https://aflow.org"
     API="/API/aflux/?"
     MATCHBOOK="species((C:Si:Ge:Sn),!Pb)"
     DIRECTIVES="$paging(1,10)"
     SUMMONS=MATCHBOOK+","+DIRECTIVES

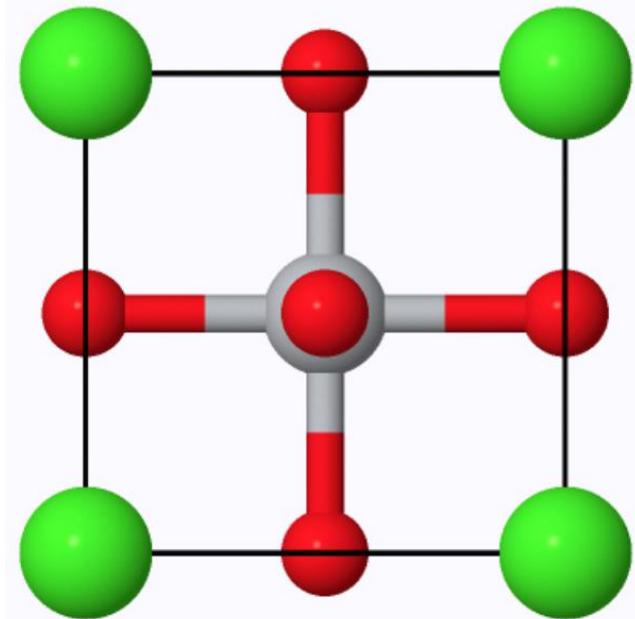
     print(SERVER+API+SUMMONS)

     response=json.loads(urlopen(SERVER+API+SUMMONS).read().decode("utf-8"))
     print(response)

https://aflow.org/API/aflux/?species\(\(C:Si:Ge:Sn\),!Pb\),\$paging\(1,10\)
[{'compound': 'Ag1Al1Au1Sn1', 'auid': 'aflow:4389c382c27f3f50', 'aurl': 'aflowlib.duke.edu:AFL0WDATA/LIB4_RAW/AgAlAuSn:PAW_PBE/ABCD_cF16_216_c_d_b_a.ABCD',
```

AFLUX: Structure Type Search

- Prototype labels:



- Search for NaCl in rocksalt structure:
prototype label is “AB_cF8_225_a_b”

```
<API>/?species(Na,Cl),nspecies(2),  
aflow_prototype_label_relax(AB_cF8_225_a_b)
```

Perovskite structure:

ABC₃_cP5_221_a_b_c

Pearson
Composition

Space group

Wyckoff positions

```
3s  import json, sys, os  
      from urllib.request import urlopen  
  
      SERVER="https://aflow.org"  
      API="/API/aflux/?"  
      MATCHBOOK="species(Na,Cl),nspecies(2),aflow_prototype_label_relax(AB_cF8_225_a_b)"  
      DIRECTIVES="$paging(0)"  
      response=json.loads(urlopen(SERVER+API+MATCHBOOK).read().decode("utf-8"))  
      print(response)  
  
      [{"compound": "Cl1Na1", "auid": "aflow:39ab5e62afdb5ac0", "aurl": "aflowlib.duke.edu:AFLOWDATA/ICSD_WEB/FCC/Cl1Na1_ICSD_622369"},
```

Mehl et al., Comput. Mater. Sci. **136**, S1-S828 (2017);
Hicks et al., Comput. Mater. Sci. **161**, S1-S1011 (2019);
Hicks et al., npj Comput. Mater. **7**, 30 (2021).

AFLUX: AFLOW Search API

- Example: use AFLOW-specific keywords to search for lead-free cubic halide perovskites with a band gap greater than 3 eV

```
<API>/?species( (F:Cl:Br:I) , !Pb) , nspecies(3) ,  
aflow_prototype_label_relax(ABC3_cP5_221_a_b_c) , Egap(3*)
```

Search for lead-free halide perovskites with band-gap greater than 3eV

```
[5] import json, sys, os  
18s   from urllib.request import urlopen  
  
 SERVER="https://aflow.org"  
 API="/API/aflux/?"  
 MATCHBOOK="species((F:Cl:Br:I),!Pb),nspecies(3),aflow_prototype_label_relax(ABC3_cP5_221_a_b_c),Egap(3*)"  
 DIRECTIVES="$paging(0)"  
 SUMMONS=MATCHBOOK+","+DIRECTIVES  
  
 response=json.loads(urlopen(SERVER+API+SUMMONS).read().decode("utf-8"))  
 print(response)  
  
[{'compound': 'Ag1Be1F3', 'auid': 'aflow:aa197e3461543939', 'aurl': 'aflowlib.duke.edu:AFLOWDATA/LIB3_RAW/AgBe_svF/T0009.ABC:LDAU2',
```

- Can also use “Halogens” to search for group of elements

```
<API>/?species(Halogens, !Pb) , nspecies(3) ,  
aflow_prototype_label_relax(ABC3_cP5_221_a_b_c) , Egap(3*)
```

AFLOW Search Page: Exercises

1. Use the AFLUX API to find the VRH bulk moduli (“ael_bulk_modulus_vrh”) for materials containing Ti in the AFLOW database. What is the material with the highest bulk modulus? What is the space group of this material? Is it a metal or an insulator (“Egap_type”)?
2. Use the AFLUX API to find the electronic band gaps (“Egap”) for all rocksalt structure (“AB_cF8_225_a_b”) alkali halide (“AlkaliMetals”, “Halogens”) materials in the AFLOW database. How many entries are returned?

Materials Project API

- Materials Project uses MPRester API
- Special client needs to be downloaded and installed (not available through Colab):
`pip install mp-api`
`from mp_api.client import MPRester`
- Materials Project uses a “document-based” database (MongoDB); data is returned as document objects
- Materials Project API also requires an API key (related to your login/registration on MP website)
- To use pip in Colab: `!pip install mp-api`

Materials Project API

- MP API key available at: <https://next-gen.materialsproject.org/api>

TABLE OF CONTENTS

[The Materials Project API](#)

[API Key](#)

[Accessing Data](#)

[Documentation](#)

[Additional APIs](#)

[Getting Help](#)

API Key

Your API Key

<your-API-key-here>

To make any request to the Materials Project API, you must use an API key. Your API key is generated for you automatically upon registering with the Materials Project website and is synced with the email you used to register.

Remember to keep your API key safe and to not share it with anyone you do not trust.

If you are logged in, you can always access your API key from this page or from your [dashboard](#).

If you intend heavy API usage, you can give us a heads up by sending a message to heavy.api.use@materialsproject.org. This is not required, but may help us if we see unusual load on our servers. For large and/or long-running queries, we ask that you please make a local copy and only retrieve the data once.

- MP API key will be alphanumeric string unique to your account on Materials Project

Materials Project API

- MPRester API usage example: search for Si-O compounds with band gap between 0.5 and 1.0 eV:

```
with MPRester("<your_api_key_here>") as mpr:
```

```
    docs = mpr.summary.search(elements=["Si", "O"], band_gap=(0.5, 1.0))
```

Compositions with Si and O

Band gap between 0.5eV and 1.0eV

- This will return all data associated with the entries satisfying this query. To restrict returned data to specific attributes, use “fields”:

```
with MPRester("<your_api_key_here>") as mpr:
```

```
    docs = mpr.summary.search(elements=["Si", "O"], band_gap=(0.5, 1.0),  
        fields=["material_id", "band_gap", "volume"])
```

Return only this data

Materials Project API

```
▶ !pip install mp-API
from mp_api.client import MPRester

with MPRester("<your-api-key-here>") as mpr:
    docs = mpr.summary.search(elements=["Si", "O"], band_gap=(0.5, 1.0), fields=["material_id", "band_gap", "volume"])
    print(docs)

[{"volume": 1013.9709061448139, "material_id": MPID(mp-1224617), "band_gap": 0.8252000000000002, "fields_not_requested": ["builder_meta", "nsites", "elements", "nelements", "composition", "composition_reduced", "formula_pretty", "formula_anonymous", "chemsys", "density", "density_atomic", "symmetry", "properties"]}, {"volume": 695.498126677145, "material_id": MPID(mp-1227528), "band_gap": 0.7827999999999999, "fields_not_requested": ["builder_meta", "nsites", "elements", "nelements", "composition", "composition_reduced", "formula_pretty", "formula_anonymous", "chemsys", "density", "density_atomic", "symmetry", "properties"]}, {"volume": 946.9207311486335, "material_id": MPID(mp-1224625), "band_gap": 0.9364999999999999, "fields_not_requested": ["builder_meta", "nsites", "elements", "nelements", "composition", "composition_reduced", "formula_pretty", "formula_anonymous", "chemsys", "density", "density_atomic", "symmetry", "properties"]}, {"volume": 1881.884702293924, "material_id": MPID(mp-686232), "band_gap": 0.5483, "fields_not_requested": ["builder_meta", "nsites", "elements", "nelements", "composition", "composition_reduced", "formula_pretty", "formula_anonymous", "chemsys", "density", "density_atomic", "symmetry", "properties"]}, {"volume": 995.8436795717986, "material_id": MPID(mp-1203093), "band_gap": 0.7426, "fields_not_requested": ["builder_meta", "nsites", "elements", "nelements", "composition", "composition_reduced", "formula_pretty", "formula_anonymous", "chemsys", "density", "density_atomic", "symmetry", "properties"]}, {"volume": 2313.151551863883, "material_id": MPID(mp-1229169), "band_gap": 0.6122000000000001, "fields_not_requested": ["builder_meta", "nsites", "elements", "nelements", "composition", "composition_reduced", "formula_pretty", "formula_anonymous", "chemsys", "density", "density_atomic", "symmetry", "properties"]}, {"volume": 957.6037554914742, "material_id": MPID(mp-1173853), "band_gap": 0.9215, "fields_not_requested": ["builder_meta", "nsites", "elements", "nelements", "composition", "composition_reduced", "formula_pretty", "formula_anonymous", "chemsys", "density", "density_atomic", "symmetry", "properties"]}]
```

Materials Project API

- Important keywords:
`elements`: chemical elements in composition
`band_gap`: electronic band gap
`material_id`: Materials Project ID for material
`volume`: cell volume
- Can also download all data for a particular entry using its MP ID:
`with MPRester("<your_api_key_here>") as mpr:`
 `data = mpr.materials.get_data_by_id("mp-22862")`
- More information: <https://docs.materialsproject.org/downloading-data/how-do-i-download-the-materials-project-database>

Optimade: Common Materials Search API

- Rapid expansion of Materials Databases and APIs creates issues with having to learn a new API for each database
- Therefore, efforts are being made to implement common APIs and standards
- Optimade Common API is an API developed by the Optimade consortium:
<https://www.optimade.org/>
- Current full Optimade specification:
<https://github.com/Materials-Consortia/OPTIMADE/blob/master/optimade.rst>

C. Anderson et al., Sci. Data **8**, 217 (2021).

Optimade: Common Materials Search API

- Databases supporting Optimade include: optimade.org/providers-dashboard/
 - AFLOW: aflow.org/API/optimade/v1
 - Alexandria: alexandria.icams.rub.de/pbe/v1/
 - Computational Materials Repository: <https://cmrdb.fysik.dtu.dk>
 - COD: Crystallography Open Database: crystallography.net/cod/optimade
 - Jarvis: jarvis.nist.gov
 - NoMaD: nomad-lab.eu/prod/rae/optimade/
 - Materials Cloud: materialscloud.org/optimade/
 - Materials Platform for Data Science: api.mpds.io
 - Materials Project: optimade.materialsproject.org
 - OQMD: oqmd.org/optimade/
- Full current list is available at: optimade.org/providers-dashboard/

Optimade: Common Materials Search API

- Optimade supports several logical operators:
 - Basic Boolean operations: “AND”, “OR”, “NOT”
 - Numeric and string comparisons: “<”, “>”, “=”, “<=”, “>=”, “!=”
 - Substring comparisons: “CONTAINS”, “STARTS WITH”, “ENDS WITH”
 - List comparisons: “HAS”, “HAS ALL”, “HAS ANY” “HAS ONLY”, “LENGTH”
- Optimade keywords:
 - “elements”: list of elements to query
 - “nelements”: number of elements
- Optimade endpoints: **structures, calculations**

C. Anderson et al., Sci. Data **8**, 217 (2021).

Optimade: Common Materials Search API

- Optimade query URI has the following structure:

```
<scheme>://<authority>/<path>/<endpoint>?filter=<query>
```

server endpoint query

- Example: Optimade API on AFLOW, search for materials containing Ti:
 - Server scheme/authority/path: <https://aflow.org/API/optimade/v1/>
 - Structures endpoint: **structures**
 - Query: **elements HAS “Ti”**
 - Full URI:
[https://aflow.org/API/optimade/v1/structures?filter=elements HAS “Ti”](https://aflow.org/API/optimade/v1/structures?filter=elements%20HAS%20%22Ti%22)

C. Anderson et al., Sci. Data 8, 217 (2021).

Optimade: Common Materials Search API

JSON Raw Data Headers

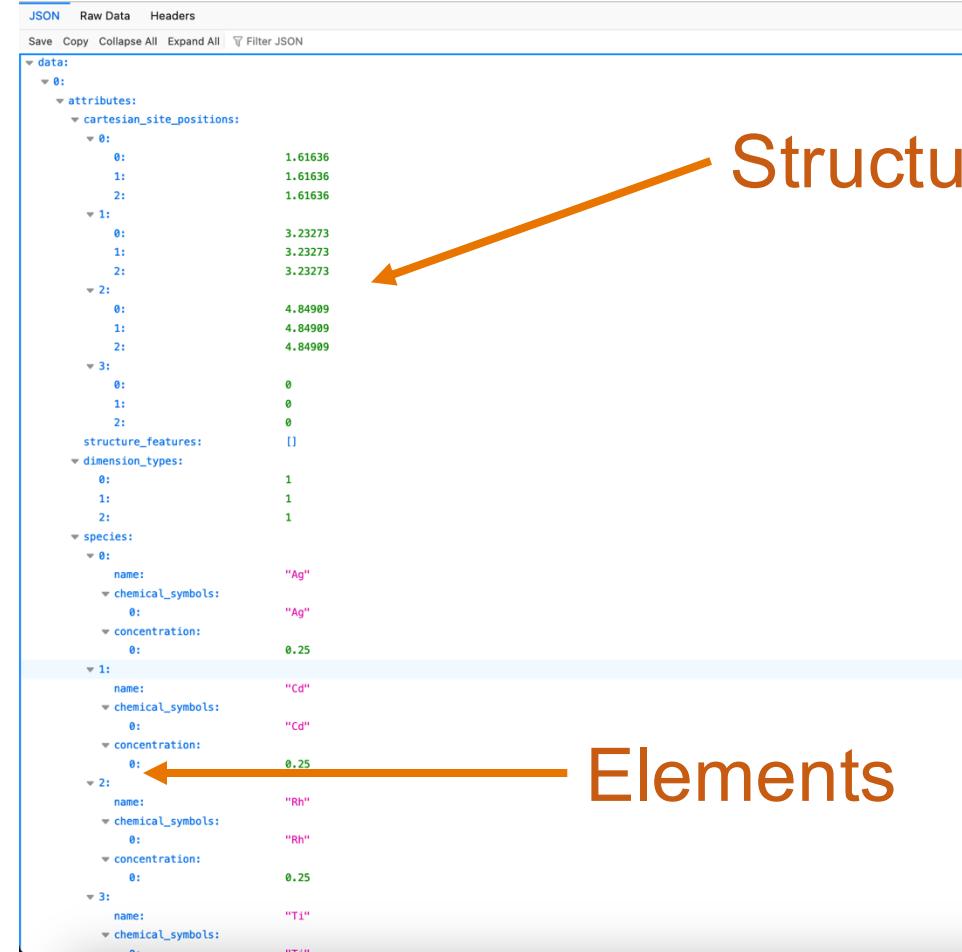
Save Copy Collapse All Expand All Filter JSON

▶ data: [...] ← Materials data

meta:
query:
representation: "/v1/structures?filter=elements%20HAS%20ONLY%20%22Ti%22"
api_version: "1.0.0"
schema: "<https://schemas.optimade.org/openapi/v1.0.0/optimade.json>"
time_stamp: "2024-02-25T01:35:40Z"
more_data_available: true
provider:
name: "AFLLOW"
description: "Automatic-FLOW (AFLLOW) Computational Materials Data Repository"
prefix: "aflow"
homepage: "<https://aflow.org>"
implementation:
name: "aflow-optimade-ng"
version: "ada0d743"
source_url: "<https://aflow.org/install-aflow>"
maintainer:
email: "optimade@aflowlib.duke.edu"
issue_tracker: "<https://groups.io/g/aflow>"
_aflux_aflux_query: "[http://aflow.org/API/aflux/?auid,aflowlib_date,composition,compound,natoms,nspecies,positions_cartesian,species,stoichiometry,paging\(1,100\),format\(json\)](http://aflow.org/API/aflux/?auid,aflowlib_date,composition,compound,natoms,nspecies,positions_cartesian,species,stoichiometry,paging(1,100),format(json))"
data_available: 3530330 ← Links
data_returned: 3530330 ← Total data available
links:
first: 'https://aflow.org/API/optimade/v1/structures?filter=elements%20HAS%20ONLY%20%22Ti%22&page_number=1&page_limit=100'
next: 'https://aflow.org/API/optimade/v1/structures?filter=elements%20HAS%20ONLY%20%22Ti%22&page_number=2&page_limit=100'

Optimade: Common Materials Search API

- Example: Optimade API on AFLOW, search for materials containing Ti:
[https://aflow.org/API/optimade/v1/structures?filter=elements HAS "Ti"](https://aflow.org/API/optimade/v1/structures?filter=elements HAS 'Ti')



The screenshot shows a JSON API response from the Optimade API on AFLOW. The response is a list of materials, each represented by a 'data' object. The 'data' object contains various attributes such as 'cartesian_site_positions', 'structure_features', 'dimension_types', and 'species'. The 'species' attribute is expanded to show individual elements with their names ('Ag', 'Cd', 'Rh', 'Ti') and chemical symbols ('Ag', 'Cd', 'Rh', 'Ti'). A red arrow points from the text 'Structural data' to the 'cartesian_site_positions' section of the JSON. Another red arrow points from the text 'Elements' to the 'species' section.

```
JSON Raw Data Headers
Save Copy Collapse All Expand All Filter JSON

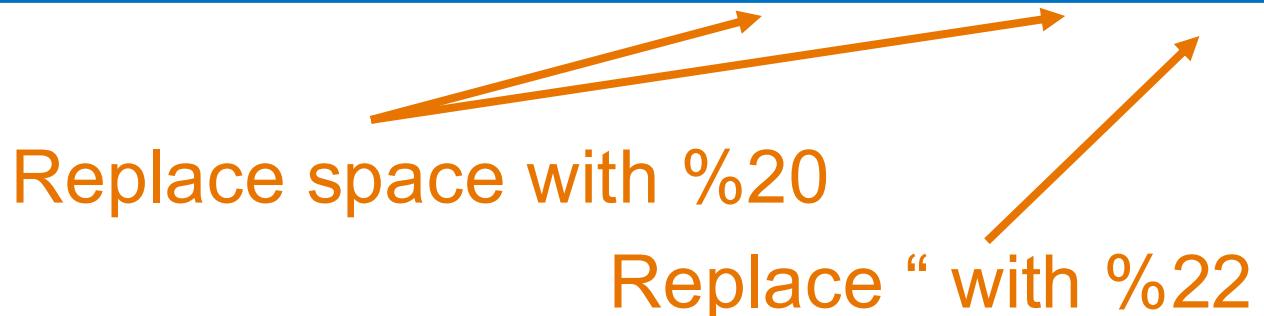
data:
  0:
    attributes:
      cartesian_site_positions:
        0:
          0: 1.61636
          1: 1.61636
          2: 1.61636
        1:
          0: 3.23273
          1: 3.23273
          2: 3.23273
        2:
          0: 4.84989
          1: 4.84989
          2: 4.84989
        3:
          0: 0
          1: 0
          2: 0
      structure_features: []
    dimension_types:
      0: 1
      1: 1
      2: 1
    species:
      0:
        name: "Ag"
        chemical_symbols:
          0: "Ag"
        concentration:
          0: 0.25
      1:
        name: "Cd"
        chemical_symbols:
          0: "Cd"
        concentration:
          0: 0.25
      2:
        name: "Rh"
        chemical_symbols:
          0: "Rh"
        concentration:
          0: 0.25
      3:
        name: "Ti"
        chemical_symbols:
          0: "Ti"
```

Structural data

Elements

Optimade: Common Materials Search API

- Optimade syntax contains reserved characters: e.g. spaces, quotation marks
- These need to be escaped with some browsers and with Python urllib library to avoid errors
- Replace space with "%20"; replace quotation marks with "%22"
- Example: Optimade API on AFLOW, search for materials containing Ti:
<https://aflow.org/API/optimade/v1/structures?filter=elements%20HAS%20%22Ti%22>



Replace space with %20
Replace " with %22

Optimade: Common Materials Search API

- Example: Using Python to query AFLOW using Optimade API, search for materials containing Ti:

[https://aflow.org/API/optimade/v1/structures/?filter=elements HAS “Ti”](https://aflow.org/API/optimade/v1/structures/?filter=elements HAS ‘Ti’)

Programmatically retrieving data using the Optimade Common API

Search AFLOW for materials with element Ti

```
[2] import json, sys, os
    from urllib.request import urlopen

    SERVER='https://aflow.org'
    API='/API/optimade/v1/structures?'
    FILTER='filter=elements%20HAS%20%22Ti%22'

    print(SERVER+API+FILTER)

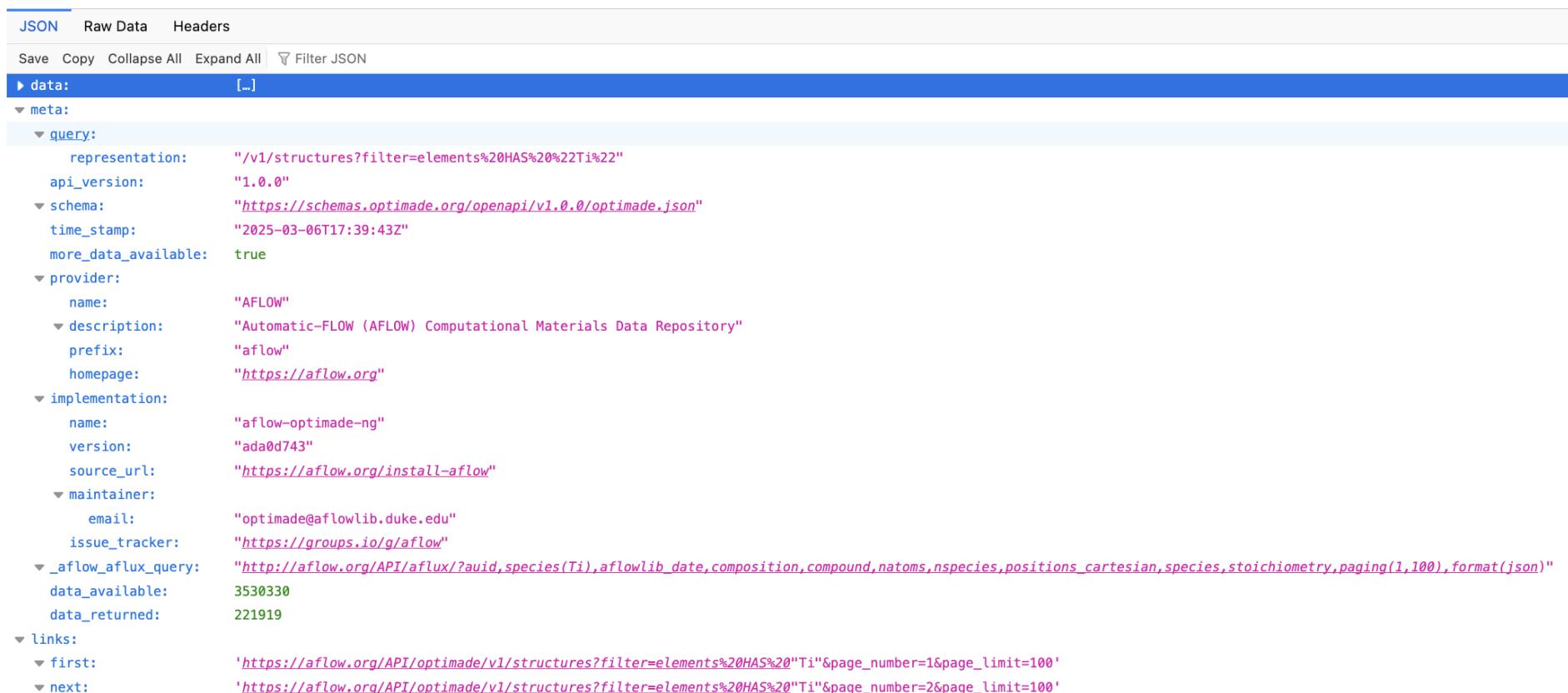
    response=json.loads(urlopen(SERVER+API+FILTER).read().decode("utf-8"))
    print(response)

https://aflow.org/API/optimade/v1/structures?filter=elements%20HAS%20%22Ti%22
{'data': [{}'attributes': {'cartesian_site_positions': [[1.61636, 1.61636, 1.61636], [3.23273, 3.23273, 3.23273], [4.84909, 4.84909, 4.84909], [0, 0, 0]]},
```

Optimade: Common Materials Search API

- Example: Using Python to query AFLOW using Optimade API, search for materials containing Ti:

[https://aflow.org/API/optimade/v1/structures/?filter=elements HAS "Ti"](https://aflow.org/API/optimade/v1/structures/?filter=elements HAS \)



The screenshot shows a JSON viewer interface with the following structure:

```
JSON Raw Data Headers
Save Copy Collapse All Expand All Filter JSON
▶ data: [...]
  ▼ meta:
    ▼ query:
      representation: "/v1/structures?filter=elements%20HAS%20%22Ti%22"
      api_version: "1.0.0"
    ▼ schema:
      https://schemas.optimade.org/openapi/v1.0.0/optimade.json
    time_stamp: "2025-03-06T17:39:43Z"
    more_data_available: true
  ▼ provider:
    name: "AFLOW"
  ▼ description:
    prefix: "aflow"
    homepage: "https://aflow.org"
  ▼ implementation:
    name: "aflow-optimade-ng"
    version: "ada0d743"
    source_url: "https://aflow.org/install-aflow"
  ▼ maintainer:
    email: "optimade@aflowlib.duke.edu"
    issue_tracker: "https://groups.io/g/aflow"
  ▼ _aflow_aflux_query:
    data_available: 3530330
    data_returned: 221919
  ▼ links:
    ▼ first:
      https://aflow.org/API/optimade/v1/structures?filter=elements%20HAS%20"Ti"&page_number=1&page_limit=100
    ▼ next:
      https://aflow.org/API/optimade/v1/structures?filter=elements%20HAS%20"Ti"&page_number=2&page_limit=100
```

Optimade: Common Materials Search API

- Example: Using Python to query NOMAD using Optimade API, search for materials containing Ti and O only:

[https://nomad-lab.eu/prod/rae/optimade/structures?filter=elements HAS ONLY
“Ti”, “O”](https://nomad-lab.eu/prod/rae/optimade/structures?filter=elements HAS ONLY %22Ti%22,%22O%22)

Search Nomad for materials containing only Ti and O

```
✓ 2s ⏪ import json, sys, os
    from urllib.request import urlopen

    SERVER='https://nomad-lab.eu'
    API='/prod/rae/optimade/structures?'
    FILTER='filter=elements%20HAS%20ONLY%20%220%22,%22Ti%22'

    print(SERVER+API+FILTER)

    response=json.loads(urlopen(SERVER+API+FILTER).read().decode("utf-8"))
    print(response)

→ https://nomad-lab.eu/prod/rae/optimade/structures?filter=elements%20HAS%20ONLY%20%220%22,%22Ti%22
{'data': [{id: 'kA1ZbAWVgKlBpBU4FzWgbglu1gbJ', type: 'structures', attributes: {immutable_id: 'kA1ZbAWVgKlBpBU4FzWgbglu1gbJ',
```

Optimade: Common Materials Search API

- Example: Using Python to query NOMAD using Optimade API, search for materials containing Ti and O only:

<https://nomad-lab.eu/prod/rae/optimade/structures?filter=elements HAS ONLY "Ti", "O">

JSON Raw Data Headers

Save Copy Collapse All Expand All (slow) Filter JSON

▶ data: [...]

▼ meta:

- ▶ query: {...}
- api_version: "1.0.1"
- more_data_available: true
- time_stamp: "2024-02-25T01:51:54Z"
- data_returned: 904
- ▶ provider: {...}
- data_available: 10000
- ▶ implementation: {...}
- included: []
- ▶ links: {...}

JSON Raw Data Headers	
Save Copy Collapse All Expand All (slow) Filter JSON	
data:	0:
id:	"kA1zbAWgK1BpBU4FzWgbglu1gbJ"
type:	"structures"
attributes:	
immutable_id:	"kA1zbAWgK1BpBU4FzWgbglu1gbJ"
last_modified:	"2022-04-25T14:08:37Z"
elements:	[...]
n_elements:	2
elements_ratios:	[...]
chemical_formula_descriptive:	"O108Ti108"
chemical_formula_reduced:	"O108Ti108"
chemical_formula_hill:	"O108Ti108"
chemical_formula_anonymous:	"A108B108"
dimension_types:	[...]
nperiodic_dimensions:	3
lattice_vectors:	[...]
cartesian_site_positions:	[...]
nsites:	216
species:	[...]
species_at_sites:	[...]
structure_features:	[]
_nmd_dft_crystal_system:	"unavailable"
_nmd_dft_quantities:	[...]
_nmd_calc_hash:	"7GGNECBYVPFg4I8mnIPPCBXZYATY"
_nmd_files:	[...]
_nmd_dft_system:	"unavailable"
_nmd_only_atoms:	"OTi"
_nmd_dft_code_name:	"FHI-aims"
_nmd_dft_spacegroup:	-1
_nmd_raw_id:	"E700D13F-9E44-44A0-AA8B-BEF975FE47F9"
_nmd_nomad_version:	"0.10.11"
_nmd_external_id:	null
_nmd_dft_n_geometries:	1
_nmd_last_edit:	null
_nmd_dft_basis_set:	"numeric_AOs"
_nmd_external_db:	null

Total data
returned for this
request

Optimade: Common Materials Search API

- Example: search AFLOW for materials containing C, Si, Ge, Sn, but not Pb:

```
<API>/?filter=elements HAS ANY "C", "Si", "Ge", "Sn" AND  
NOT elements HAS "Pb"
```

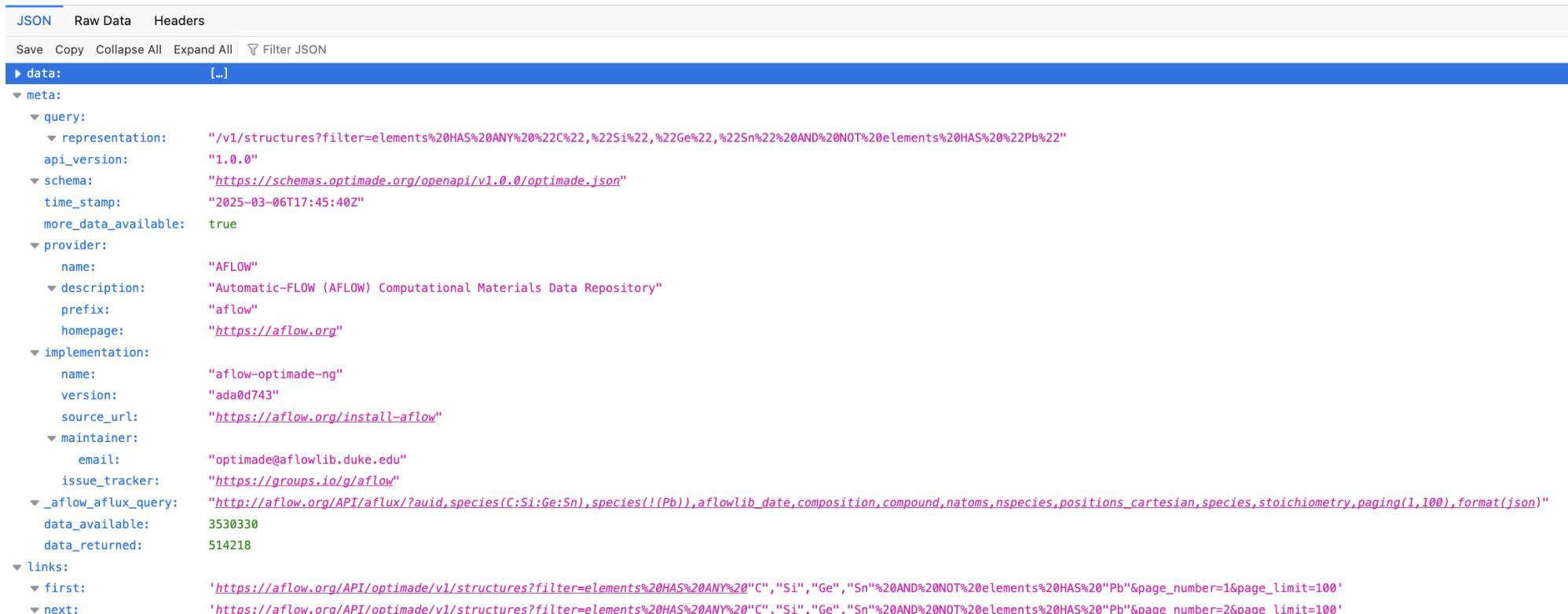
Search AFLOW for materials with C or Si or Ge or Sn but not Pb

```
✓ 41s  play  
import json, sys, os  
from urllib.request import urlopen  
  
SERVER='https://aflow.org'  
API='/API/optimade/v1/structures?'  
FILTER='filter=elements%20HAS%20ANY%20%22C%22,%22Si%22,%22Ge%22,%22Sn%22%20AND%20NOT%20elements%20HAS%20%22Pb%22'  
  
print(SERVER+API+FILTER)  
  
response=json.loads(urlopen(SERVER+API+FILTER).read().decode("utf-8"))  
print(response)  
  
https://aflow.org/API/optimade/v1/structures?filter=elements%20HAS%20ANY%20%22C%22,%22Si%22,%22Ge%22,%22Sn%22%20AND%20NOT%20elements%20HAS%20%22Pb%22  
{'links': {'base_url': {'href': 'http://aflow.org/API/optimade'}, 'meta': {'_aflow_db_version': '1.0'}}, '_aflow_base_url':
```

Optimade: Common Materials Search API

- Example: search AFLOW for materials containing C, Si, Ge, Sn, but not Pb:

```
<API>/?filter=elements HAS ANY "C", "Si", "Ge", "Sn" AND  
NOT elements HAS "Pb"
```



The screenshot shows a JSON viewer interface with the following structure:

- JSON Raw Data Headers**: Buttons at the top.
- Save Copy Collapse All Expand All Filter JSON**: Tools for managing the JSON data.
- data: []**: The main data object.
- meta:** A nested object containing metadata about the search query.
 - query:** The search query string: "/v1/structures?filter=elements%20HAS%20ANY%20%22C%22,%22Si%22,%22Ge%22,%22Sn%22%20AND%20NOT%20elements%20HAS%20%22Pb%22".
 - api_version:** "1.0.0".
 - schema:** "<https://schemas.optimade.org/openapi/v1.0.0/optimade.json>".
 - time_stamp:** "2025-03-06T17:45:40Z".
 - more_data_available:** true.
- provider:** Information about the provider.
 - name:** "AFLOW".
 - description:** "Automatic-FLOW (AFLOW) Computational Materials Data Repository".
 - prefix:** "aflow".
 - homepage:** "<https://aflow.org>".
- implementation:** Information about the implementation.
 - name:** "aflow-optimade-ng".
 - version:** "ada0d743".
 - source_url:** "<https://aflow.org/install-aflow>".
- maintainer:** Information about the maintainer.
 - email:** "optimade@aflowlib.duke.edu".
 - issue_tracker:** "<https://groups.io/g/aflow>".
- _aflux_aflux_query:** The search query string: "[http://aflux.org/API/aflux/?auid,species\(C:Si:Ge:Sn\),species\(!\(Pb\)\),aflowlib_date,composition,compound,natoms,nspecies,positions_cartesian,species,stoichiometry,paging\(1,100\),format\(json\)](http://aflow.org/API/aflux/?auid,species(C:Si:Ge:Sn),species(!(Pb)),aflowlib_date,composition,compound,natoms,nspecies,positions_cartesian,species,stoichiometry,paging(1,100),format(json))".
- data_available:** 3530330.
- data_returned:** 514218.
- links:** Links to the first and next pages of results.
 - first:** "[https://aflow.org/API/optimade/v1/structures?filter=elements%20HAS%20ANY%20%C","%20Si","%20Ge","%20Sn%20AND%20NOT%20elements%20HAS%20%22Pb%22&page_number=1&page_limit=100](https://aflow.org/API/optimade/v1/structures?filter=elements%20HAS%20ANY%20%C)".
 - next:** "[https://aflow.org/API/optimade/v1/structures?filter=elements%20HAS%20ANY%20%C","%20Si","%20Ge","%20Sn%20AND%20NOT%20elements%20HAS%20%22Pb%22&page_number=2&page_limit=100](https://aflow.org/API/optimade/v1/structures?filter=elements%20HAS%20ANY%20%C)".

Optimade: Common Materials Search API

- Database-specific keywords can also be used to search for properties in that database
- In Optimade, a database-specific prefix is appended to the database-specific keyword
- This is something that is still under development for the databases supporting the Optimade API
- Example: bulk modulus in AFLOW:
 - Keyword for band gap AFLOW is “ael_bulk_modulus_vrh”
 - The prefix for AFLOW is “_aflow_”
 - In AFLOW-OPTIMADE, this becomes: “_aflow_ael_bulk_modulus_vrh”

```
<API>/?filter=elements HAS "Ti" AND  
_aflow_ael_bulk_modulus_vrh > 100.0
```

Optimade: Common Materials Search API

- Pagination in Optimade can be performed using the following keywords:
 - page_limit: number of entries per page
 - page_number: number of page
 - page_offset: number of entries to skip
- Optimade pagination commands come after query "?" but before filter:

```
<API>/structures?page_offset=100&page_limit=50&filter=elements HAS Ti
```

C. Anderson et al., Sci. Data **8**, 217 (2021).

AFLOW Search Page: Exercises

1. Use the Optimade API to find binary ($n\text{elements}=2$) materials containing Al in the AFLOW database. How many entries are returned?
2. Use the Optimade API to find materials containing Li but not Na in the NOMAD database. How many entries are returned?
3. Use the Optimade API to find ternary materials containing Nb in the NOMAD database. How many entries are returned?

Conclusions

- Computational and experimental materials data is becoming widely available online
- Data can be retrieved both through web portals as well as programmatically using REST-APIs
- Several databases have their own search APIs, enabling the programmatic submission of complex search queries and data retrieval
- Optimade Common API enables access to several materials databases, including aflow.org, materialsproject.org, nomad-lab.eu, oqmd.org, materialscloud.org, etc.

Acknowledgements

- Frisco Rose, Marco Esters, Eric Gossett, Corey Oses, David Hicks, Simon Divilov, Hagen Eckert, Stefano Curtarolo (Duke): AFLOW database and AFLUX search API
- Matthew Evans, Gian-Marco Rignanese (UCLouvain), Casper Anderson (EPFL), Rickard Armiento (Linköping University) et al. : Optimade Common API
- Patrick Huck, Anubhav Jain, Kristin Persson et al. (LBNL): Materials Project
- Markus Scheidgen, Claudia Draxl et al. (Humboldt Universität zu Berlin): NOMAD



AFLOW
Automatic - FLOW for Materials Discovery

