The Open Reaction Database (ORD) initiative for standardizing and sharing organic reaction data

Connor W. Coley
Assistant Professor
MIT Chemical Engineering

NIH Virtual Workshop on InChI
March 23, 2021

Governance and Acknowledgements

Governing Committee
- Connor Coley (MIT)
- Abby Doyle (Princeton, C-CAS)
- Spencer Dreher (Merck)
- Joel Hawkins (Pfizer)
- Klavs Jensen (MIT)
- Steven Kearnes (Google)

Early contributors / volunteers
- Anton Kast (Google)
- Michael Maser (Caltech)
- Nathan Kim (MIT)
- Michael Wleklinski (Merck)

Advisory Board
- Juan Alvarez (Merck)
- Alán Aspuru-Guzik (Toronto, MADNESS)
- Tim Cernak (Michigan)
- Lucy Colwell (Cambridge, SynTech, Google)
- Werngard Czechtizky (AstraZeneca)
- Matthew Gaunt (Cambridge, SynTech)
- Mimi Hii (Imperial, ROAR)
- Greg Landrum (T5 Informatics)
- Fabio Lima (Novartis)
- Christos Nicolaou (Lilly)
- Sarah Reisman (Caltech)
- Matthew Sigman (Utah, C-CAS)
- Jay Stevens (BMS)
- Sarah Trice (Entos)
- Matt Tudge (GSK)

Overview

Goals and use cases

Design considerations

From the documentation: "support machine learning and related efforts in reaction prediction, chemical synthesis planning, and experiment design”

Goals:

- Provide a structured data format for chemical reaction data
- Provide an interface for easy browsing and downloading of data
- Make reaction data freely and publicly available for anyone to use
- Encourage sharing of precompetitive proprietary data, especially HTE data

Primary use cases: synthetic organic chemistry

1. High-throughput experimentation
   a. Data are recorded in spreadsheet formats including only varied parameters;
   b. One template Reaction is defined to specify all aspects held constant;
   c. The Dataset is defined by iterating over the spreadsheet and creating one Reaction entry per experimental condition.

2. “Traditional” bench chemistry
   a. A chemist uses a graphical webform to define the settings and outcomes of all reactions used within a paper or project;
   b. The structured Dataset is saved, uploaded to the Open Reaction Database, and used as part of their supporting information;
   c. A list of reactions is exported from the Dataset in an SI-like text format.

Applications

Example downstream ML uses

Yield or selectivity prediction

REPORT

Predicting reaction performance in C–N cross-coupling using machine learning

Derek T. Ahneman

+ See all authors and a

Science 13 Apr 2018:
Vol. 360, Issue 6385, p
DOI: 10.1126/science.

RESEARCH ARTICLE

Prediction of higher-selectivity catalysts by computer-driven workflow and machine learning

Andrew F. Zahrt*, Je

+ See all authors and affilia

Science 18 Jan 2019:
Vol. 363, Issue 6424, eaau5
DOI: 10.1126/science.aau5

Article | Published: 17 July 2019

Holistic prediction of enantioselectivity in asymmetric catalysis

Jolene P. Reid & Matthew S. Sigman

Nature **571**, 343–348(2019) | Cite this article

Multi-step retrosynthetic planning

Published: 29 March 2018

Planning chemical syntheses with deep neural

Marwin H. S

Nature 555

RESEARCH ARTICLE

A robotic platform for flow synthesis of organic compounds informed by AI planning

Connor W

+ See all aut

Science 09 A
Vol. 365, Issu
DOI: 10.1126

Software | Open Access | Published: 17 November 2020

AiZynthFinder: a fast, robust and flexible open-source software for retrosynthetic planning

Samuel Genheden, Amol Thakkar, Veronika Chadimová, Jean-Louis Reymond, Ola Engkvist & Esben Bjerrum

Journal of Cheminformatics **12**, Article number: 70 (2020) | Cite this article

Reaction condition recommendation

Using Machine Learning To Predict Suitable Conditions for Organic Reactions

Hanyu Gao, Thomas J. Struble, Connor W. Coley, Yuran Wang, William H. Green, and Klavs F. Jens

Cite this: *ACS*
Publication Date
https://doi.org/1
Copyright © 201
RIGHTS & PERMIS

Deoxyfluorination with Sulfonyl Fluorides: Navigating Reaction Space with Machine Learning

Matthew K. Nie

Cite this: *J. Am.* (
5008
Publication Date: Ma
https://doi.org/10.1
Copyright © 2018 A
RIGHTS & PERMISSIO

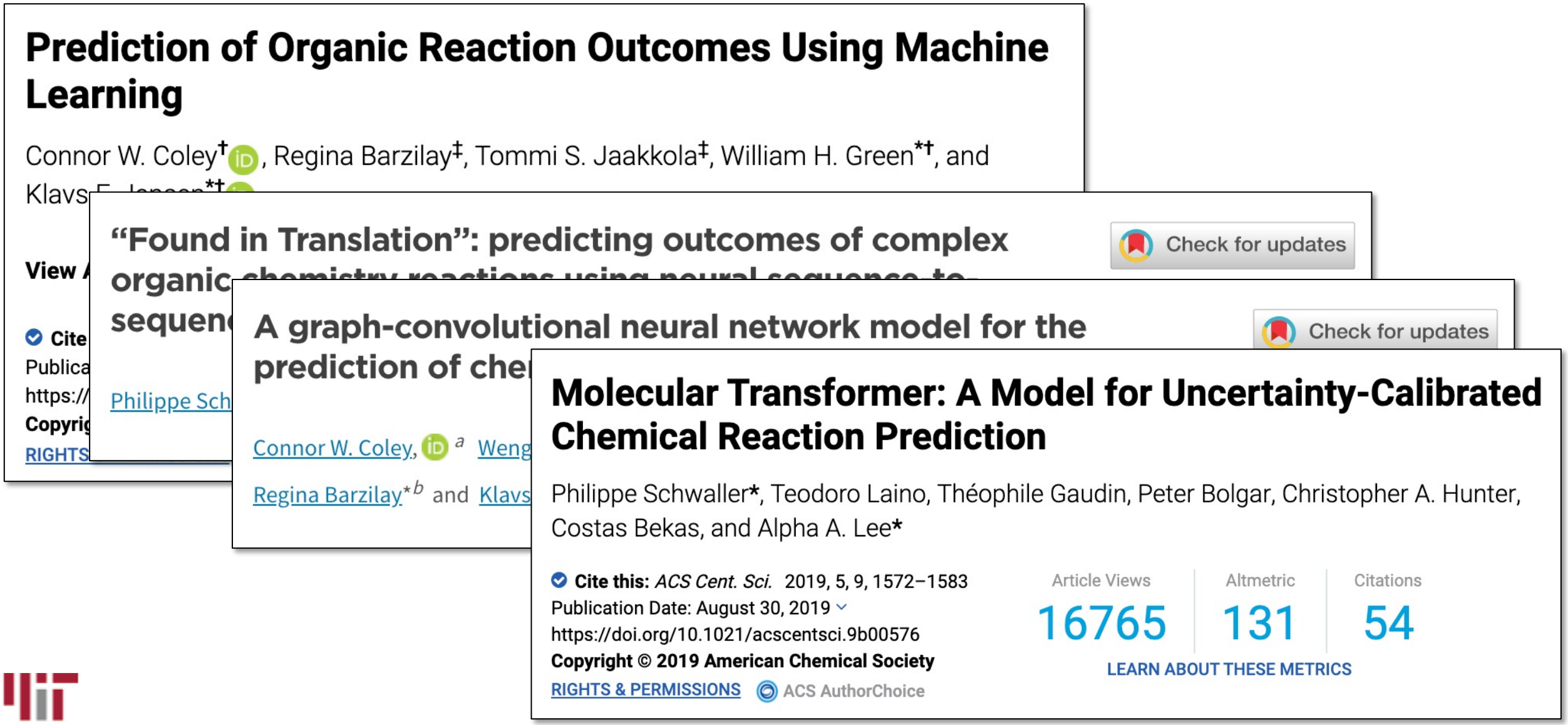
Multi-Label Classification Models for the Prediction of Cross-Coupling Reaction Conditions

Cite | Download all (5.54 MB) | Export as PDF | Share | Embed

1963	355	0
views	downloads	citations

Preprint submitted on 14.10.2020, 01:21 and posted on 15.10.2020, 06:11 by Michael Maser, Alexander Cui, Serim Ryou, Travis DeLano, Yisong Yue, Sarah Reisman

Reaction product prediction



Culture shift and new applications

We/I hope to create a culture shift in how data is shared in chemistry

- Providing a structured alternative to describing data in a .docx or .pdf (e.g., for Supporting Information documents)
  - We can't consider subtle aspects of reaction planning yet, like order of addition
- Including negative results when publishing, not just positive results
  - None of the reaction outcome prediction models can predict "no reaction"
- Releasing reaction data that does not have to be associated with a journal publication
  - Not all HTE datasets are done toward a specific publication goal
- Creating a venue that, in time, may find community traction like the PDB/CSD have
  - The chemistry community has not yet taken ownership over their own data curation
- Distributing the cost of improving our collective knowledge of reactivity
  - Individual groups/companies shouldn't need to duplicate efforts

Schema

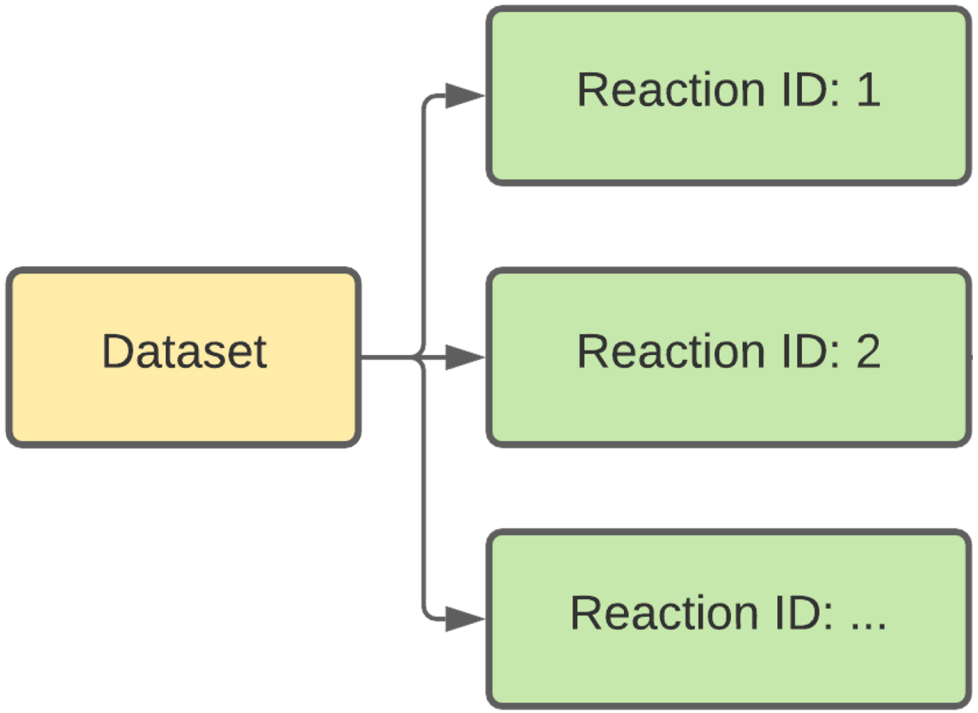
Defining the structure of reaction data

Goals for the schema

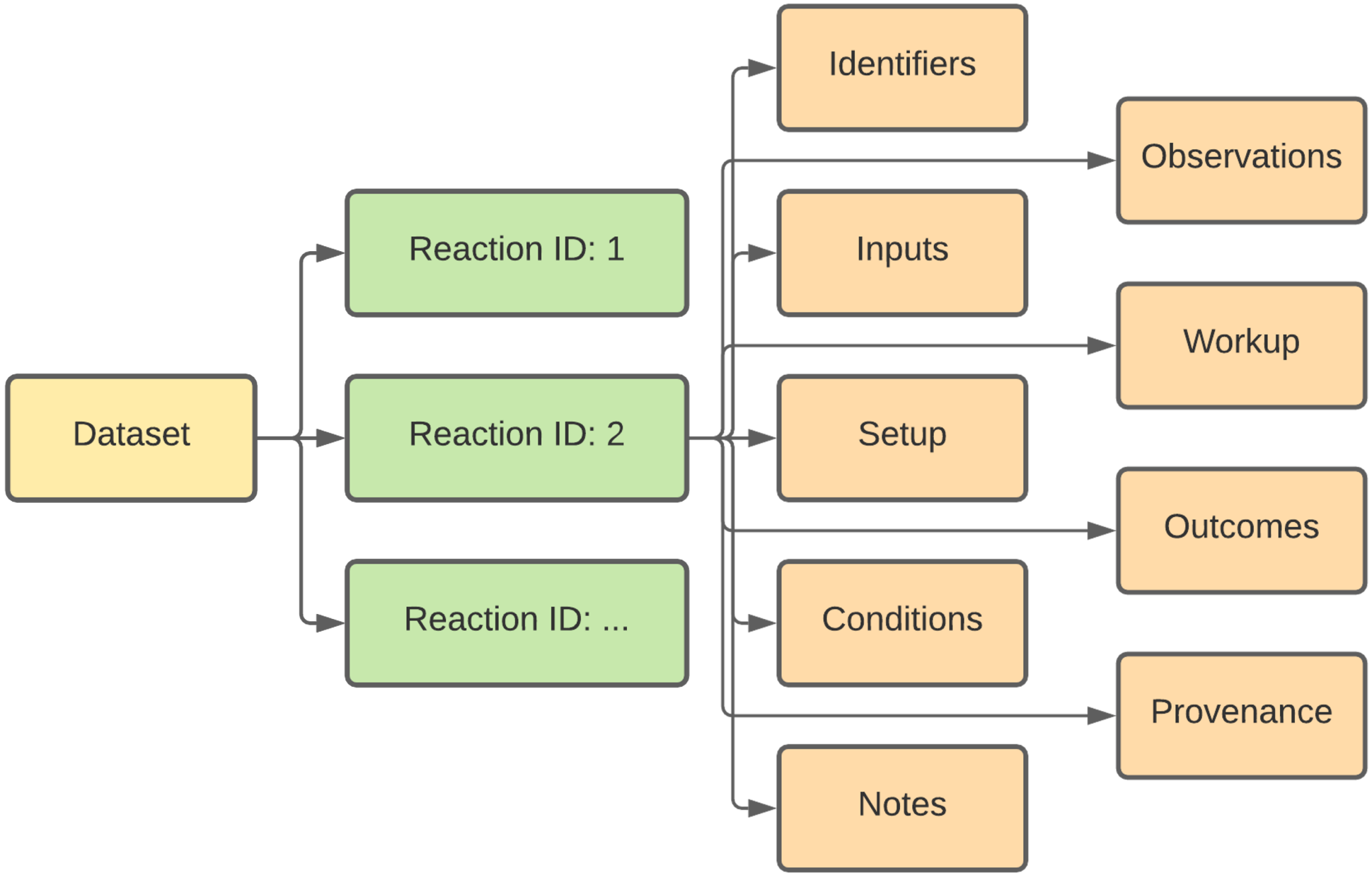
- Capture the most important aspects of reactions in a *structured* format
  - Guided by our survey Fall 2019, the focus is on single-step batch reactions
  - Fields are a superset of what existing databasing efforts contain
  - Structured data enables downstream ML applications
- Allow additional details in a flexible, *unstructured* format
- **Match chemist expectations around structure and nomenclature**
- Record what physically occurred in a chemical reaction; de-emphasize recording of a chemist's intent
  - e.g., record the actual masses and volumes that were used to create a stock solution, not the target concentration
- Be human readable

Structure of the schema

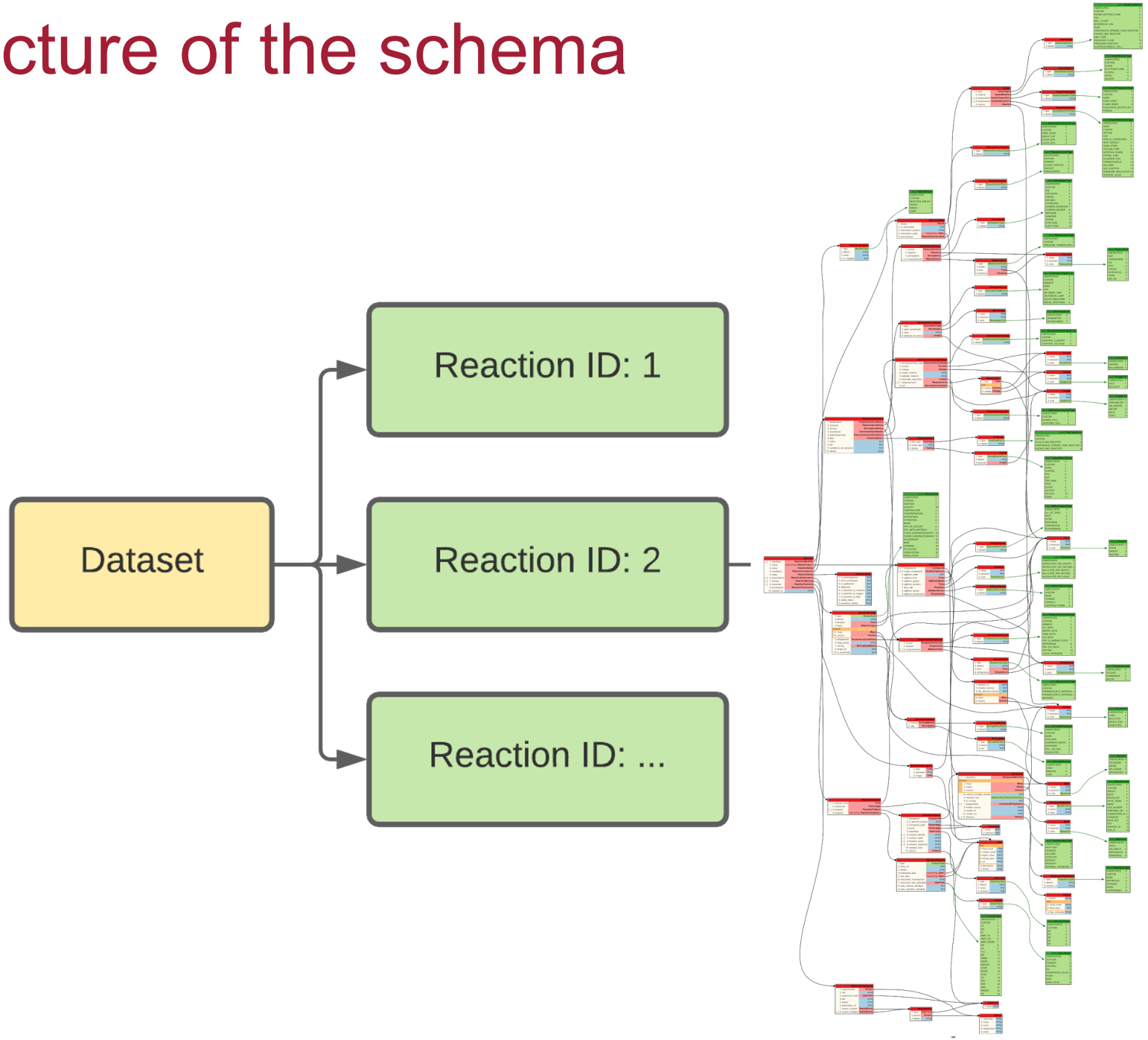
Structure of the schema



Structure of the schema



Structure of the schema



Protocol buffers

```
message Mass {
  enum MassUnit {
    UNSPECIFIED = 0;
    KILOGRAM = 1;
    GRAM = 2;
    MILLIGRAM = 3;
    MICROGRAM = 4;
  }
  float value = 1;
  // Precision of the measurement (with the same units as `value`).
  float precision = 2;
  MassUnit units = 3;
}
```

Protocol buffers

```
mass = schema.Mass(value=1.25, units='GRAM')


resolver = units.UnitResolver()
mass = resolver.resolve('1.25 g')


mass_json = """{
  "value": 1.25,
  "units": "GRAM"
}"""
mass = json_format.Parse(mass_json, schema.Mass)
```

Protocol buffers

```
reaction = schema.Reaction()
reaction.identifiers.add(value=r'deoxyfluorination', type='NAME')

# Input 1a is a stock solution of alcohol in THF
reaction.inputs['alcohol in THF'].addition_order = 1
solute = reaction.inputs['alcohol in THF'].components.add()
solvent = reaction.inputs['alcohol in THF'].components.add()

solute.reaction_role = schema.Compound.ReactionRole.REACTANT
solute.identifiers.add(value=r'c1ccccc1CCC(O)C', type='SMILES')
solute.amount.moles.CopyFrom(unit_resolver.resolve('0.1 mmol'))
solute.is_limiting = True

solvent.reaction_role = schema.Compound.ReactionRole.SOLVENT
solvent.identifiers.add(value=r'THF', type='NAME')
solvent.identifiers.add(value=r'C1CCCO1', type='SMILES')
solvent.amount.volume.CopyFrom(unit_resolver.resolve('125 uL'))
solvent.preparations.add(type='DRIED')
...
```

Enumeration procedure

1. Create a template reaction
2. Mark the variable fields in the template
3. Prepare the accompanying spreadsheet
4. Enumerate the dataset

	A	B	C	D	E
1	alcohol_smiles	sulfonyl_fluoride_smiles	base_smiles	product_smiles	product_yield
2	c1ccccc1CCC(O)C	Clc1ccc(S(=O)(=O)F)cc1	N\2=C1\N(CCCCC1)CCC/2	c1ccccc1CCC(F)C	40
3	c1ccccc1CCC(O)C	Clc1ccc(S(=O)(=O)F)cc1	CN1CCCN2C1=NCCC2	c1ccccc1CCC(F)C	54
4	c1ccccc1CCC(O)C	Clc1ccc(S(=O)(=O)F)cc1	CC(C)(C)N=C(N(C)C)N(C)C	c1ccccc1CCC(F)C	41
5	c1ccccc1CCC(O)C	Clc1ccc(S(=O)(=O)F)cc1	CC(C)(C)N=P(N1CCCC1)(N2C	c1ccccc1CCC(F)C	42
6	c1ccccc1CCC(O)C	O=S(C1=CC=CC=N1)(F)=O	N\2=C1\N(CCCCC1)CCC/2	c1ccccc1CCC(F)C	57
7	c1ccccc1CCC(O)C	O=S(C1=CC=CC=N1)(F)=O	CN1CCCN2C1=NCCC2	c1ccccc1CCC(F)C	59
8	c1ccccc1CCC(O)C	O=S(C1=CC=CC=N1)(F)=O	CC(C)(C)N=C(N(C)C)N(C)C	c1ccccc1CCC(F)C	49
9	c1ccccc1CCC(O)C	O=S(C1=CC=CC=N1)(F)=O	CC(C)(C)N=P(N1CCCC1)(N2C	c1ccccc1CCC(F)C	53
10	c1ccccc1CCC(O)C	O=S(C1=CC=C(C(F)(F)F)C=C1)	N\2=C1\N(CCCCC1)CCC/2	c1ccccc1CCC(F)C	52
11	c1ccccc1CCC(O)C	O=S(C1=CC=C(C(F)(F)F)C=C1)	CN1CCCN2C1=NCCC2	c1ccccc1CCC(F)C	69
12	c1ccccc1CCC(O)C	O=S(C1=CC=C(C(F)(F)F)C=C1)	CC(C)(C)N=C(N(C)C)N(C)C	c1ccccc1CCC(F)C	57
13	c1ccccc1CCC(O)C	O=S(C1=CC=C(C(F)(F)F)C=C1)	CC(C)(C)N=P(N1CCCC1)(N2C	c1ccccc1CCC(F)C	60
14	c1ccccc1CCC(O)C	O=S(C1=CC=C([N+]([O-])=O)(	N\2=C1\N(CCCCC1)CCC/2	c1ccccc1CCC(F)C	54
15	c1ccccc1CCC(O)C	O=S(C1=CC=C([N+]([O-])=O)(	CN1CCCN2C1=NCCC2	c1ccccc1CCC(F)C	63

```
inputs {
  key: "alcohol in THF"
  value {
    components {
      identifiers {
        type: SMILES
        value: "$alcohol_smiles$"
      }
      amount {
        moles {
          value: 0.1
          units: MILLIMOLE
        }
      }
      reaction_role: REACTANT
      is_limiting: true
    }
. . .
```

Example notebooks

Open in Colab

Deoxyfluorination with Sulfonyl Fluorides: Navigating Reaction Space with Machine Learning

DOI: 10.1021/jacs.8b01523

J. Am. Chem. Soc. 2018, 140, 5004–5008

Defining protos for reaction data in Figure 1

Colab set-up: install schema

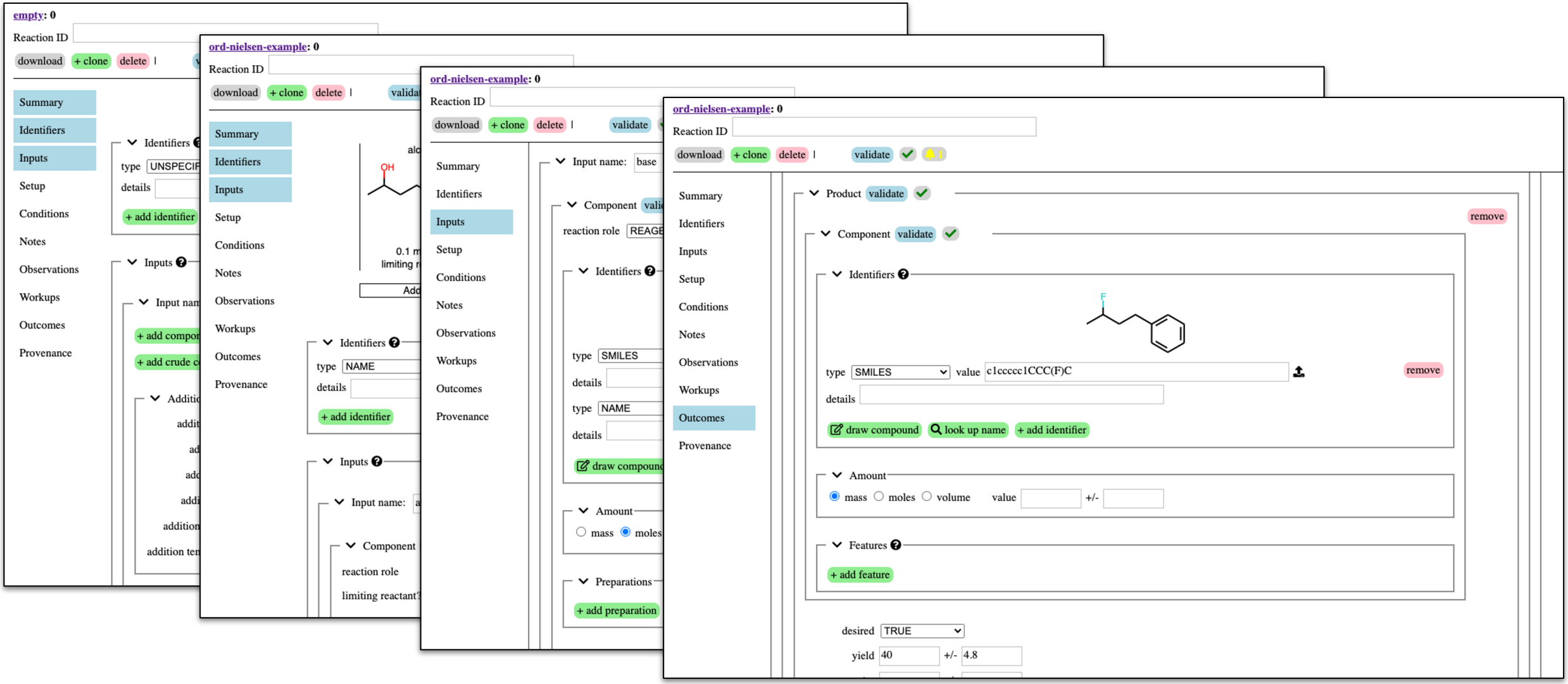
In [1]:
```
try:
    import ord_schema
    import rdkit
except:
    import sys
    !wget -c https://repo.continuum.io/miniconda/Miniconda3-latest-Linux-x86_64.sh
    !time bash ./Miniconda3-latest-Linux-x86_64.sh -b -f -p /usr/local
    !time conda install -q -y -c rdkit rdkit
    !time conda install -q -y -c anaconda protobuf
    !git clone https://github.com/Open-Reaction-Database/ord-schema.git
    %cd ord-schema
    !python setup.py install
    sys.path.append('/usr/local/lib/python3.7/site-packages/')
```

Import schema and helper functions

In [2]:
```
from datetime import datetime
from ord_schema.proto import reaction_pb2
from ord_schema.proto import dataset_pb2
from ord_schema.units import UnitResolver
from ord_schema import validations
from ord_schema import message_helpers

unit_resolver = UnitResolver()
```

Web interface at editor.open-reaction-database.org



* Looks a little different now

Where's InChI?

Opportunities for integration

Defining reactions and compounds

2

```
message ReactionIdentifier {
  // Possible identifier types are listed in an enum for extensibility
  enum IdentifierType {
    UNSPECIFIED = 0;
    CUSTOM = 1;
    REACTION_SMILES = 2;
    REACTION_CXSMILES = 6;  // Extended SMILES.
    RDFILE = 3;  // Reaction data file.
    RINCHI = 4;  // Reaction InChI.
    NAME = 5;  // Named reaction or reaction category.
  }
  IdentifierType type = 1;
  string details = 2;
  string value = 3;
  // Whether identifier contains atom-to-atom mapping information. When True,
  // we encourage users to specify how that mapping was obtained in the
  // details field (e.g., manually, using NameRXN, using ChemDraw).
  optional bool is_mapped = 4;
}
```

3

Drawing with Ketcher (native output is MolBlock)

MolFromMolBlock → MolToInChi

MolBlockToInchI?

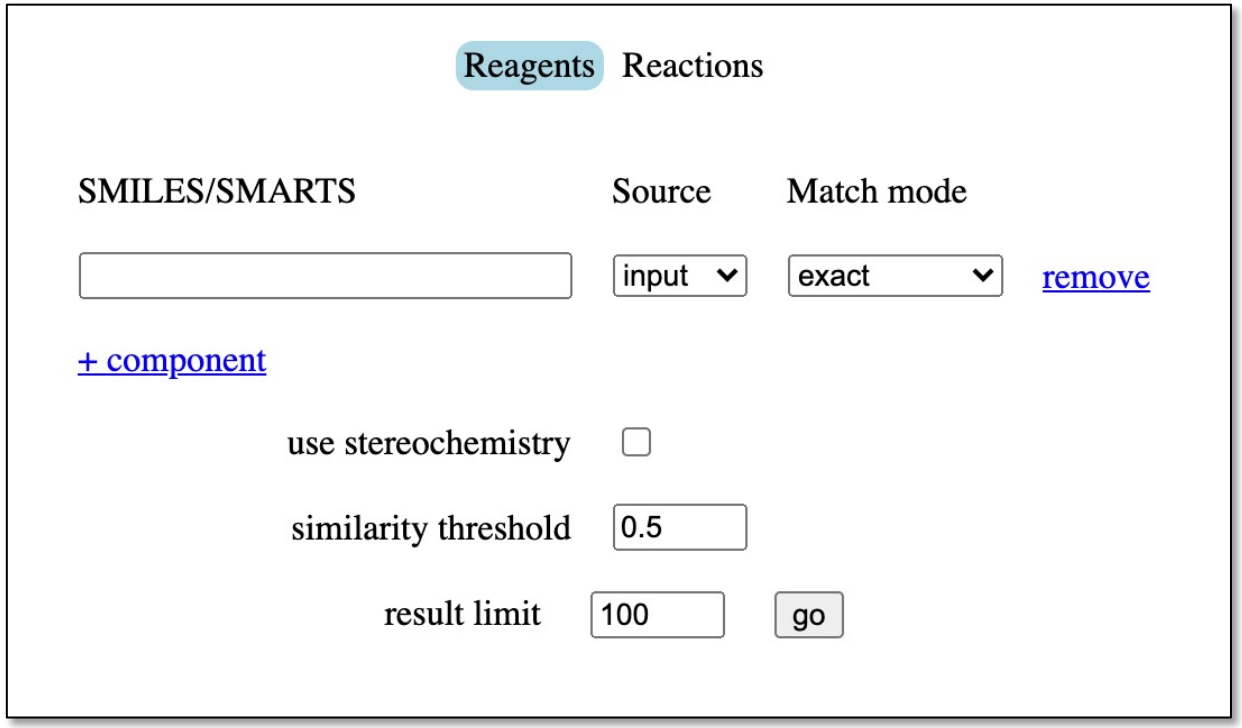
1

```
message CompoundIdentifier {
  enum IdentifierType {
    UNSPECIFIED = 0;
    CUSTOM = 1;
    // Simplified molecular-input line-entry system.
    SMILES = 2;
    // IUPAC International Chemical Identifier.
    INCHI = 3;
    // Molblock from a MDL Molfile V3000.
    MOLBLOCK = 4;
    // Chemical name following IUPAC nomenclature recommendations.
    IUPAC_NAME = 5;
    // Any accepted common name, trade name, etc.
    NAME = 6;
    // Chemical Abstracts Service Registry Number (with hyphens).
    CAS_NUMBER = 7;
    // PubChem Compound ID number.
    PUBCHEM_CID = 8;
    // ChemSpider ID number.
    CHEMSPIDER_ID = 9;
    // ChemAxon extended SMILES
    CXSMILES = 10;
    // IUPAC International Chemical Identifier key
    INCHI_KEY = 11;
    // XYZ molecule file
    XYZ = 12;
    // UniProt ID (for enzymes)
    UNIPROT_ID = 13;
    // Protein data bank ID (for enzymes)
    PDB_ID = 14;
  }
  IdentifierType type = 1;
  string details = 2;
  // Value of the compound identifier; certain types (e.g., PUBCHEM_CID) may
  // cast the string as an integer for downstream processing and validation.
  string value = 3;
}
```

Bad!

Storing and retrieving structures and reactions

1 **Reaction/substance searching (right now SMILES/SMARTS, using the RDKit PostgreSQL cartridge)**



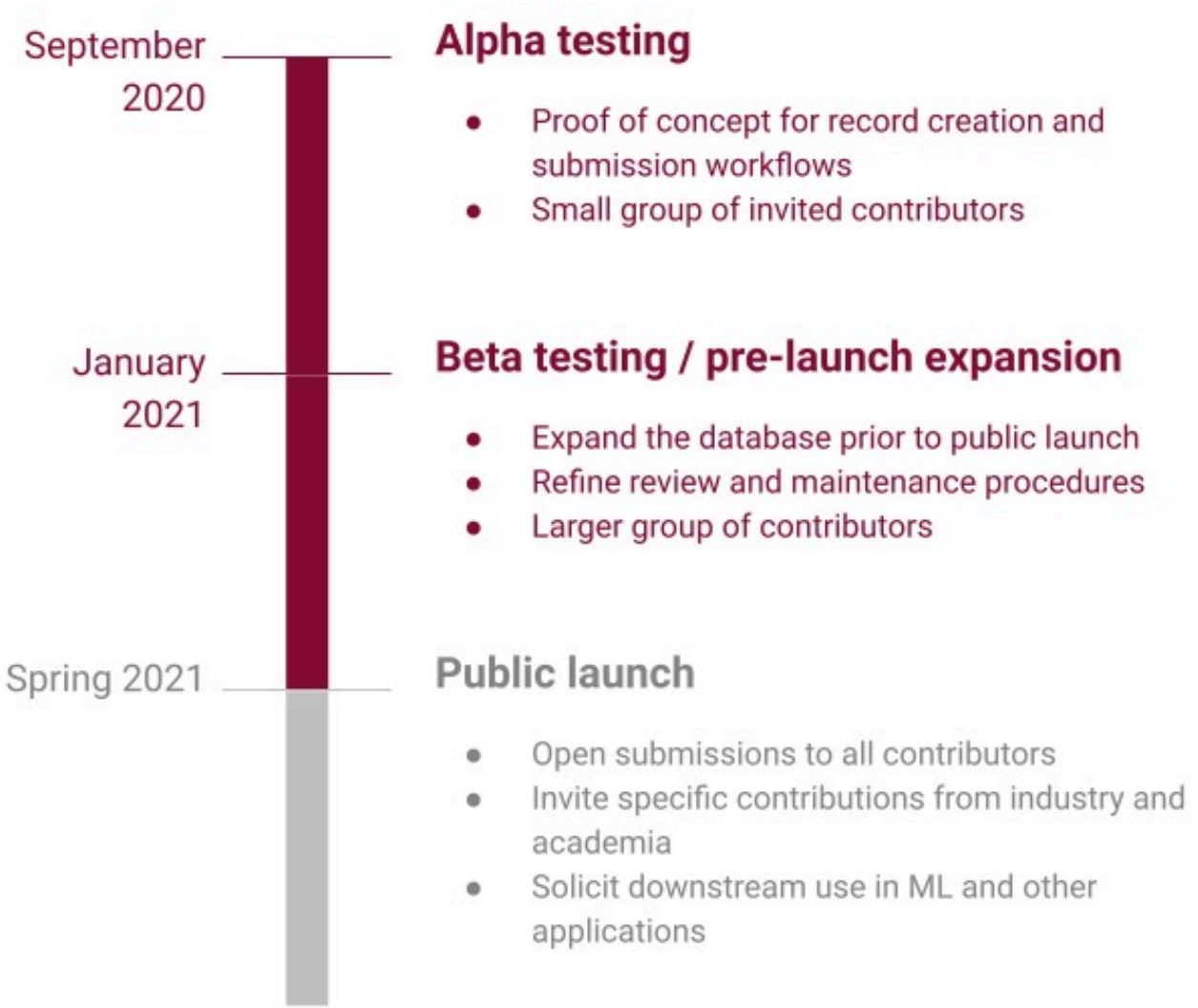
2 **Canonicalization to deduplicate**

- Different users may define compounds in very different ways
  - Drawing (Ketcher)
  - Copy/paste SMILES from ChemDraw
  - Programmatically, with SMILES, InChI, or CAS numbers
- Certain species are harder than others to capture
  - Organometallics (e.g., catalyst/ligand complexes)
  - Tautomers
  - Ionic interactions (RONa) v. (RO^-, Na^+)
- Similar challenges to what Greg mentioned for a compound registration system

Conclusion

Roadmap and summary

Development roadmap



Open Reaction Database https://open-reaction-database.org/

- Multi-institution initiative to "support machine learning and related efforts in reaction prediction, chemical synthesis planning, and experiment design” by creating a schema for organic reaction data & establishing an open access repository
- Everything is being done in the open on GitHub
- Tutorials are on YouTube
- **Currently in beta testing and looking for volunteers who can contribute data (previously published or otherwise)!**

Contact: ccoley@mit.edu