# Chemical mixtures: File format, open source tools, example data, and mixtures InChI derivative

**Alex M. Clark**

CDD VAULT®
Complexity Simplified

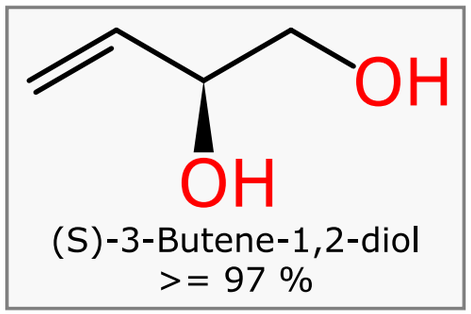
# Premise

- Representing specific chemicals routine since 1980s
  - e.g. the MDL Molfile CTAB for organics
- Most real world encounters are *mixtures*
- No industry standard format...
  - ... even though it's rather easy
  - interchange is done with text

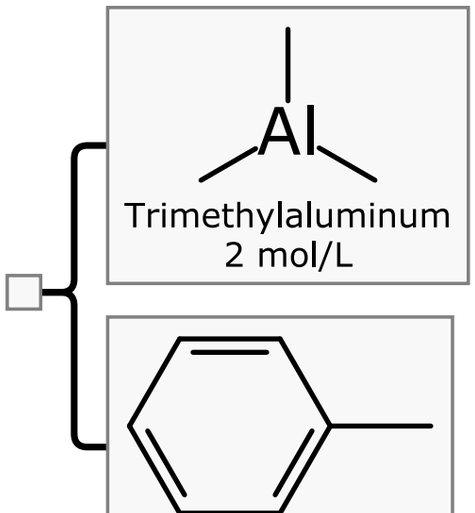
# Goal

- Grant-supported project to:
  - define a simple format (***Mixfile***)
  - create open source tools for editing & manipulating
  - bootstrap content via text mining
  - work with IUPAC for interoperability (***MInChI***)
- Results pubished in *Journal of Cheminformatics* **11**:33

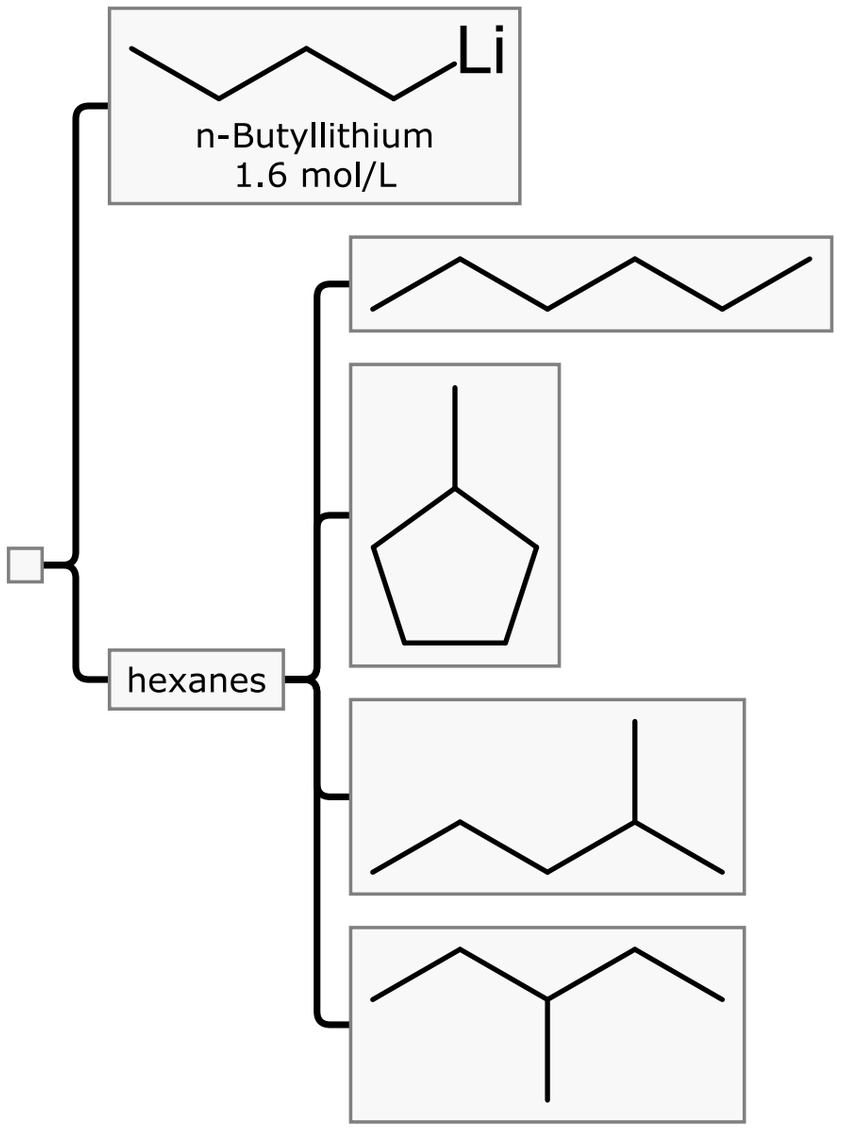
# Common Lab Mixtures




impurity




simple solution




multisolvent solution




isomers

# Real World Mixtures

## drug tablet

Eletriptan Tablet

- eletriptan hydrobromide 20 mg
- excipients
  - croscarmellose sodium 5 mg CASRN 74811-65-7
  - sodium starch glycolate 7.6 mg CASRN 9063-38-1
  - D-mannitol 50 mg CASRN 69-65-8
  - magnesium stearate 2.4 mg CASRN 557-04-0
  - talc 1.5 mg CASRN 14807-96-6
  - microcrystalline cellulose 63.5 mg CASRN 9004-34-6

## toothpaste

toothpaste

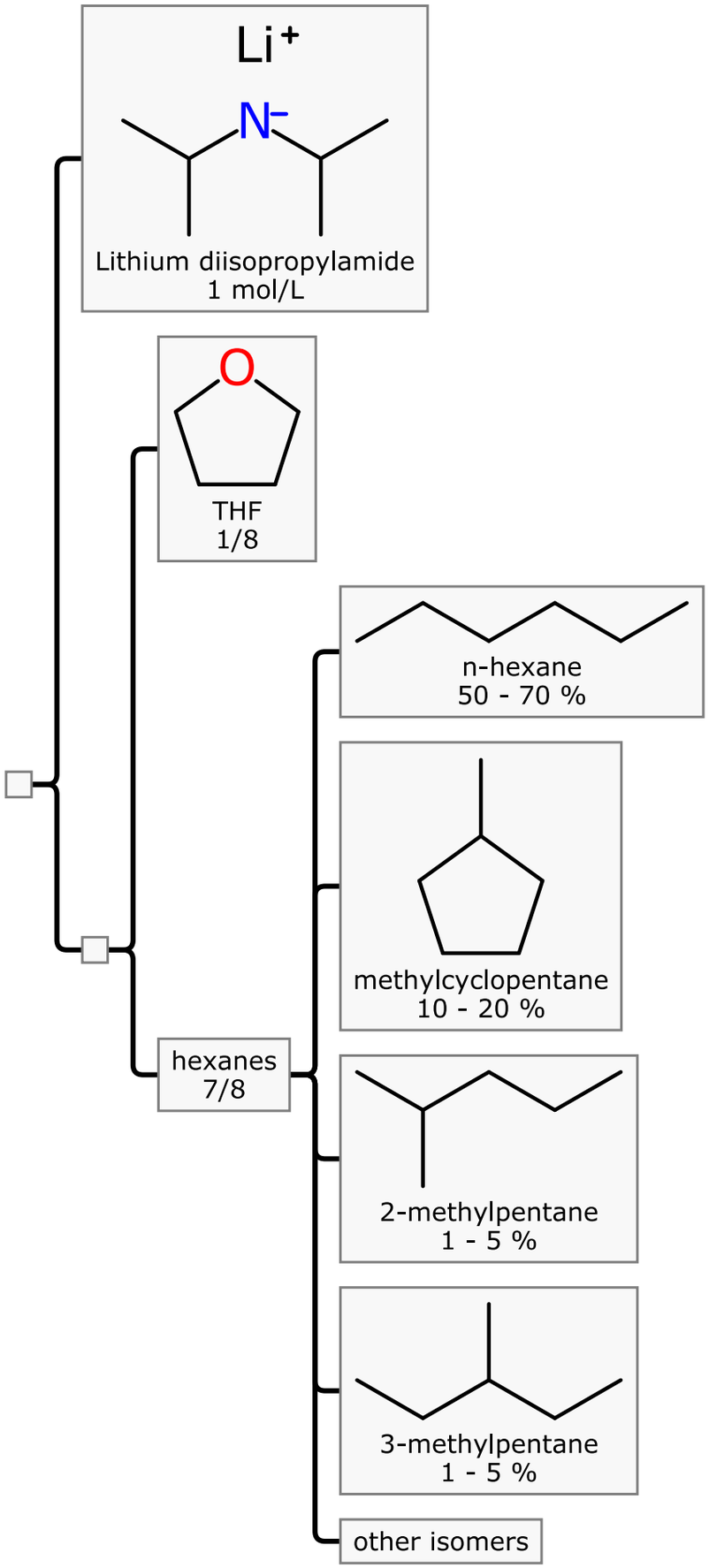
- sodium fluoride 0.243 w/v%
- inactive ingredients
  - water
  - sorbitol
  - hydrated silica
  - disodium pyrophosphate
  - sodium lauryl sulfate
  - sodium hydroxide
  - sodium saccharin
  - titanium dioxide
  - blue 1
  - cellulose gum CASRN 9004-32-4
  - carbomer CASRN 9003-01-4
  - mica CASRN 12001-26-2
  - flavor

# File Format

- *Mixfile* is to mixtures as *Molfile* is to molecules

- **Components**:
  - structure & name
  - concentration
  - identifiers

- **Hierarchy**
  - captures nuances of mixing
  - relative concentrations/uncertainty



# Representation

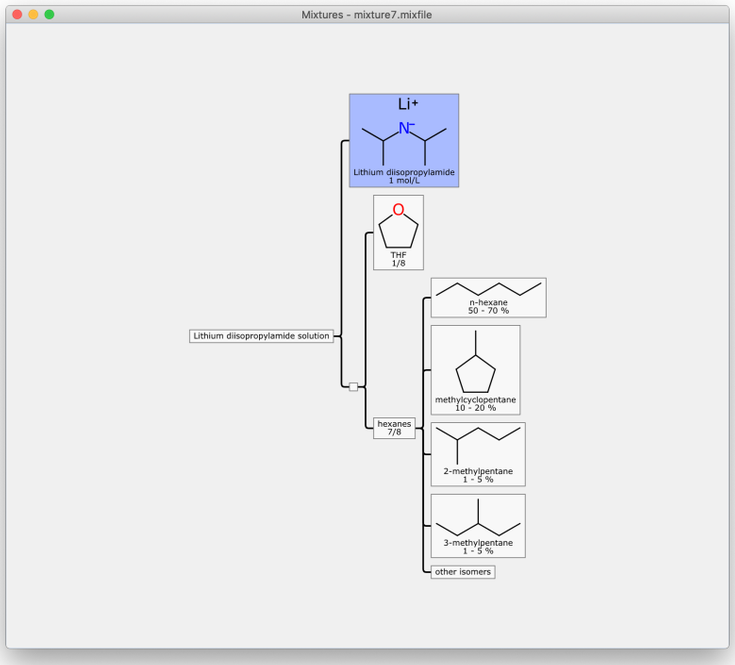
- Serialised using JSON: natural datastructure, human readable, concise, easy to code on any platform


```
{
    "mixfileVersion": 0.01,
    "name": "Lithium diisopropylamide solution",
    "contents":
    [
        {
            "name": "Lithium diisopropylamide",
            "molfile": "\nGenerated by WebMolKit\n\n  8  6  0  0  0  0  0  0  0  0999 V2000\n    0.2500    1.5000    0.0000 N   0  5  0  0  0  0  0  0  0
                       0  0  0\n   -1.0490    0.7500    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0\n   -2.3481    1.5000    0.0000 C   0  0  0  0  0
                       0  0  0  0  0  0  0\n   -1.0490   -0.7500    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0\n    1.5490    0.7500    0.0000 C   0
                       0  0  0  0  0  0  0  0  0  0  0\n    2.8481    1.5000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0\n    1.5490   -0.7500
                       0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0\n    0.2500    3.0000    0.0000 Li  0  3  0  0  0  0  0  0  0  0  0  0\n  1  2  1
                       0  0  0  0\n  2  3  1  0  0  0  0\n  2  4  1  0  0  0  0\n  1  5  1  0  0  0  0\n  5  6  1  0  0  0  0\n  5  7  1  0  0  0  0\nM
                       CHG  2   1  -1   8   1\nM  END",
            "quantity": 1,
            "units": "mol/L",
            "inchi": "InChI=1S/C6H14N.Li/c1-5(2)7-6(3)4;/h5-6H,1-4H3;/q-1;+1",
            "inchiKey": "InChIKey=ZCSHNCUQKCANBX-UHFFFAOYSA-N"
        },
        {
            "contents":
            [
                {
                    "name": "THF",
                    "molfile": "\nGenerated by WebMolKit\n\n  5  5  0  0  0  0  0  0  0  0999 V2000\n   -0.2500    3.7500    0.0000 O   0  0  0  0  0  0
                        0  0  0  0  0  0\n   -1.4600    2.8700    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0\n   -1.0000    1.4400    0.0000 C   0  0
                        0  0  0  0  0  0  0  0  0  0\n    0.9600    2.8700    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0\n    0.5000    1.4400
                        0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0\n  3  2  1  0  0  0  0\n  2  1  1  0  0  0  0\n  1  4  1  0  0  0  0\n  4  5  1  0
                        0  0  0\n  5  3  1  0  0  0  0\nM  END",
                    "ratio":
                    [
                        1,
                        8
                    ],
                    "inchi": "InChI=1S/C4H8O/c1-2-4-5-3-1/h1-4H2",
                    "inchiKey": "InChIKey=WYURNTSHIVDZCO-UHFFFAOYSA-N"
                },
```

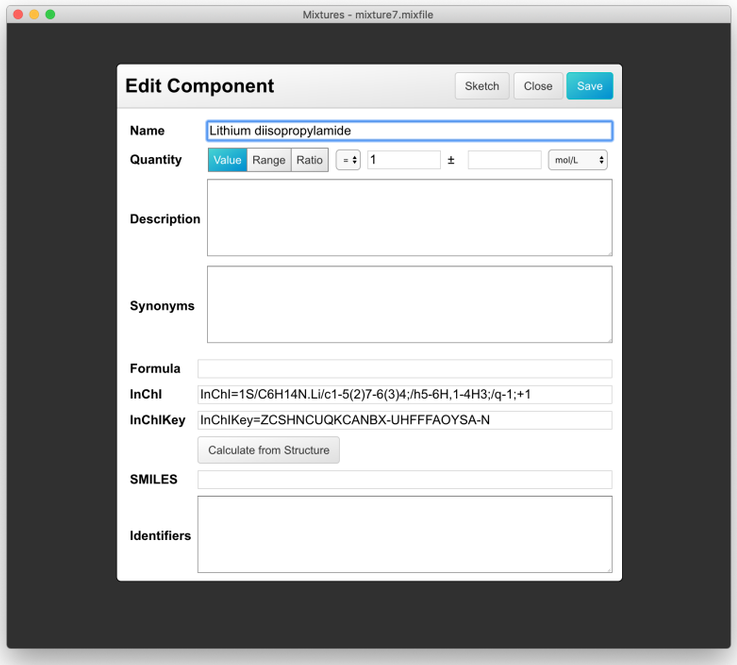
# Tooling

- Editor & libraries written in TypeScript, cheminformatics library: WebMolKit
- Create, modify, view & render *Mixfiles*
- Open source `https://github.com/cdd/mixtures`

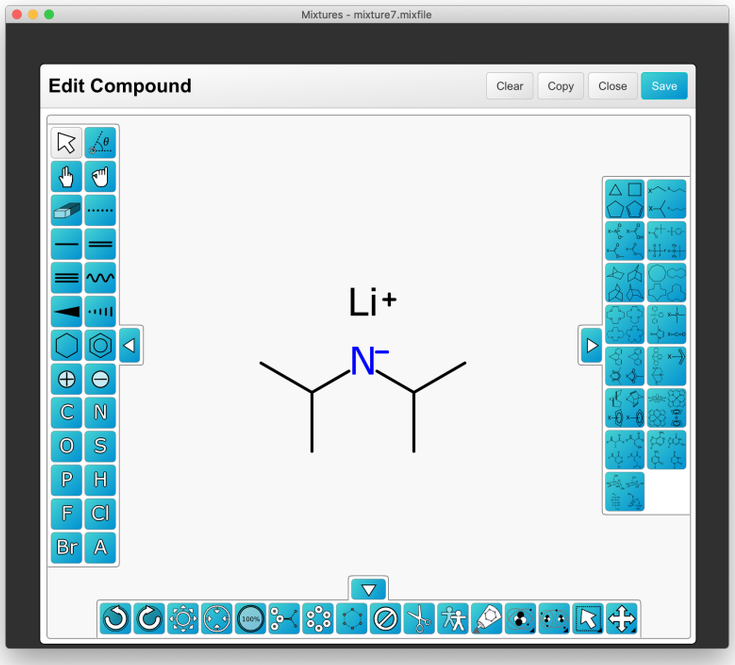
(a)




(b)

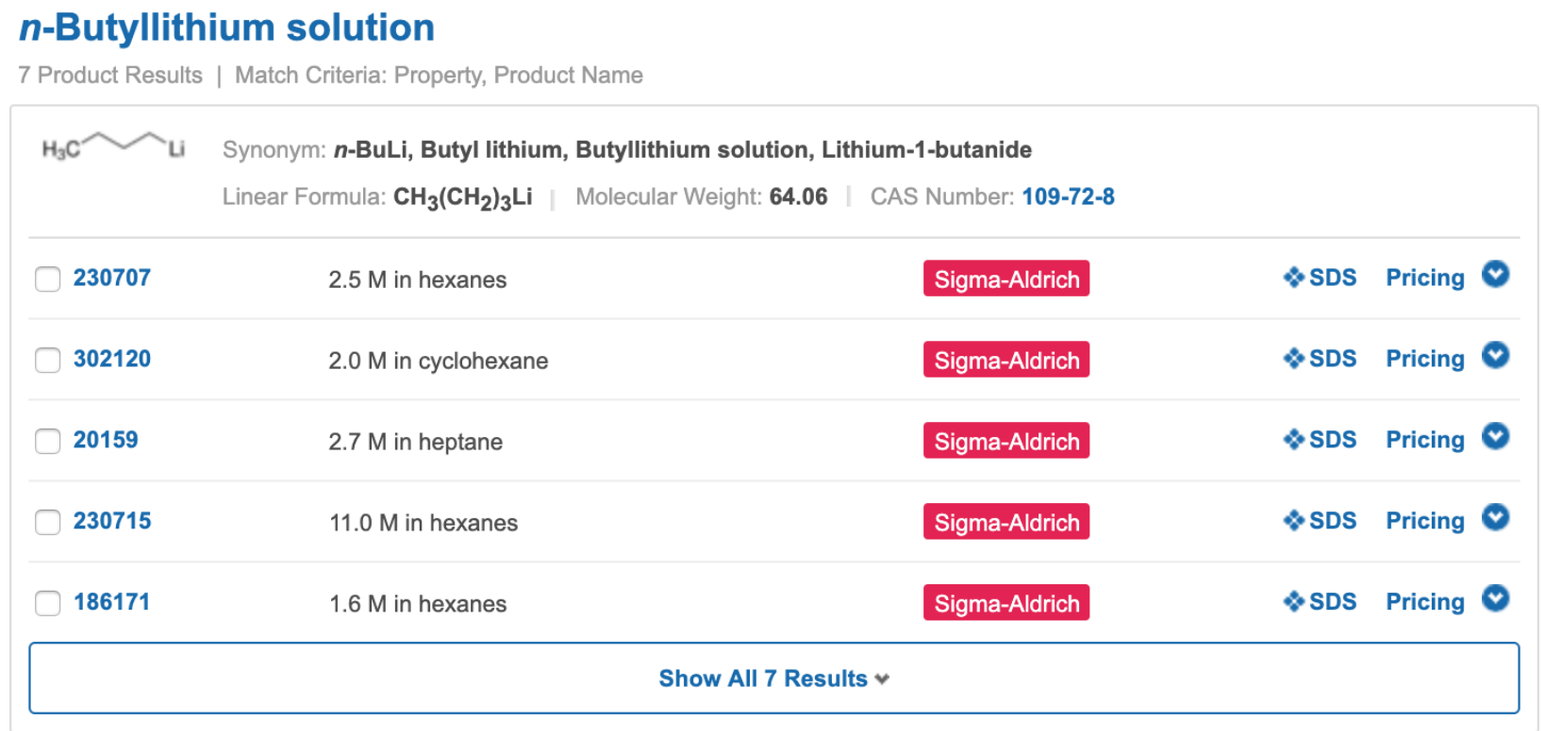



(c)



# Mixture Data

- Lots of mixtures, but mostly text
- Active ingredient often separated
  - purity
  - solvent/conc.
- Semi-structured databases also



***n*-Butyllithium solution**

7 Product Results | Match Criteria: Property, Product Name



Synonym: ***n*-BuLi, Butyl lithium, Butyllithium solution, Lithium-1-butanide**

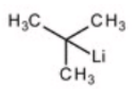
Linear Formula: $CH_3(CH_2)_3Li$ | Molecular Weight: **64.06** | CAS Number: **109-72-8**

| | | | |
|---|---|---|---|
| 230707 | 2.5 M in hexanes | Sigma-Aldrich | SDS Pricing |
| 302120 | 2.0 M in cyclohexane | Sigma-Aldrich | SDS Pricing |
| 20159 | 2.7 M in heptane | Sigma-Aldrich | SDS Pricing |
| 230715 | 11.0 M in hexanes | Sigma-Aldrich | SDS Pricing |
| 186171 | 1.6 M in hexanes | Sigma-Aldrich | SDS Pricing |

Show All 7 Results

**tert-Butyllithium**

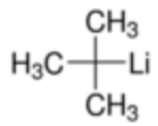
1 Product Result | Match Criteria: Product Name



| | | | |
|---|---|---|---|
| 8.14147 | (approx.15% solution in n-pentane) for synthesis | Sigma-Aldrich | SDS Pricing |

***tert*-Butyllithium solution**

1 Product Result | Match Criteria: Keyword



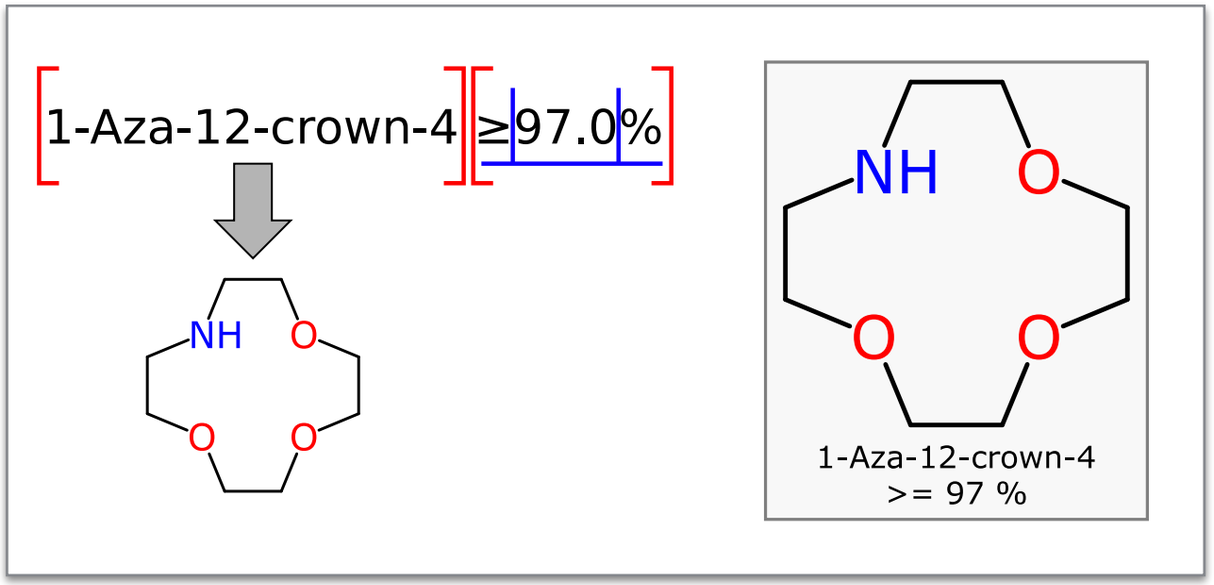
Synonym: **Lithium-2-methyl-2-propanide, t-BuLi**

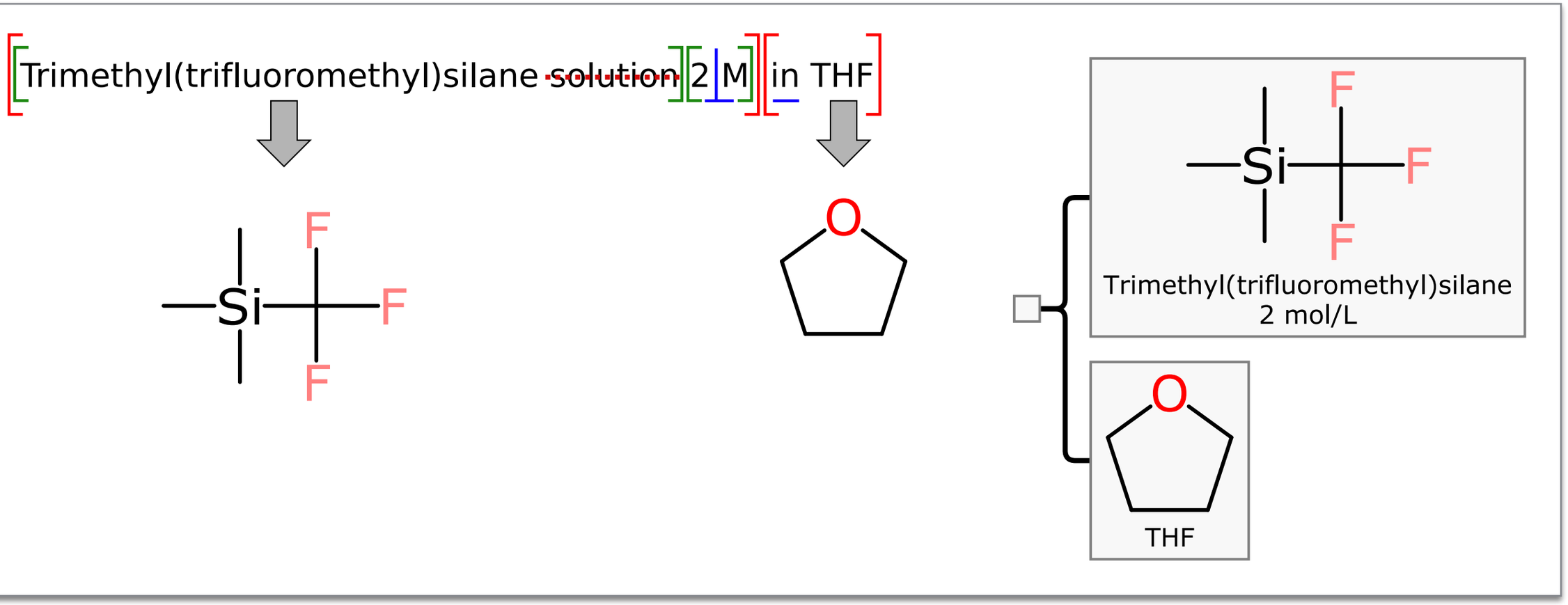
Linear Formula: $(CH_3)_3CLi$ | Molecular Weight: **64.06** | CAS Number: **594-19-4**

| | | | |
|---|---|---|---|
| 186198 | 1.6 M in pentane | Sigma-Aldrich | SDS Pricing |

# Text Extraction

- Many common patterns in text descriptions



# Brute Force

- Compose a set of regular expression rules
- Remove brand names:

```
{"effect": "remove", "regex": "(.*)A[cC][rR][oO][sS] Organics?™?(.*?)$"},
{"effect": "remove", "regex": "(.*)AcroSeal™?(.*?)$"},
{"effect": "remove", "regex": "(.*)Alfa Aesar™?(.*?)$"},
{"effect": "remove", "regex": "(.*)BioReagents™?(.*?)$"},
{"effect": "remove", "regex": "(.*)Burdick \\& Jackson™?(.*?)$"},
```

- Remove suffixes:

```
{"effect": "remove", "regex": "(.*)\\(Technical\\)$"},
{"effect": "remove", "regex": "(.*)\\(Certified\\)$"},
{"effect": "remove", "regex": "(.*)\\, pure$"},
{"effect": "remove", "regex": "(.*)\\, for analysis$"},
{"effect": "remove", "regex": "(.*)\\, extra pure$"},
```

# Brute Force (ctd)

- Concentration suffixes:

```
{"effect": "conc", "regex": "(.*) (\\d[\\d.]*)%, ee \\d[\\d.]*.%$",
   "quantity": "$2","units": "%", "relation": ""},
{"effect": "conc", "regex": "(.*) ≥ ?(\\d[\\d.]*)%$",
  "quantity": "$2","units": "%", "relation": ">="},
{"effect": "conc", "regex": "(.*) [Ss]olution (\\d[\\d.]*) ?M$",
  "quantity": "$2", "units": "mol/L"},
```

- Identify branches (with quantities):

```
{"effect": "branch", "regex":
    "(.*) \\(over (molecular sieve .*)\\)", "substance": "$2"},
{"effect": "branch", "regex": "(.*) (\\d[\\d\\.]*) ? mM in (.*)",
  "quantity": "$2", "units": "mmol/L", "substance": "$3"},
{"effect": "branch", "regex":
    "(.*) \\~?\\d[\\d\\.]*\\s?% in (.*) \\((\\~?)(\\d[\\d\\.]*) M\\)$",
    "quantity": "$4", "units": "mol/L", "relation": "$3", "substance": "$2"},
{"effect": "branch", "regex":
    "(.*)\\, (\\d[\\d\\.]*)\\s?M [Ss]olution in (.*)",
    "quantity": "$2", "units": "mol/L", "substance": "$3"},
```

# Implementation

- Apply regular expression rules greedily/recursively
  - package up into hierarchical components
  - substances referenced by *name*
- Use OPSIN for name-to-SMILES, RDKit to depict
- Use lookup database for:
  - common exceptions (e.g. trivial names)
  - named mixtures (e.g. *hexanes*)

# Results

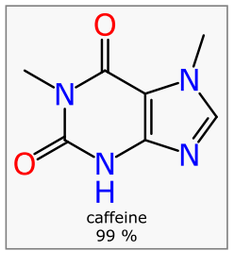
- Aggregated thousands of single-line text entries, mostly from online catalogs
- 5600 mixtures verified & included on GitHub using the *Mixfile* format
- Success rate ~95% with:
  - 250 regular expression based rules
  - 280 name lookup mappings
- All of them with calculated *MInChI* strings...

# MInChI

- Mixtures InChI is a composite notation...
  - *Mixfile* encapsulates *Molfiles* + extra metadata
  - *MInChI* encapsulates *InChI* + extra metadata
- Distilled down to the basics:
  - canonical structures
  - simplified concentration
  - nesting hierarchy
- Useful to accompany primary originated data

# Dissection

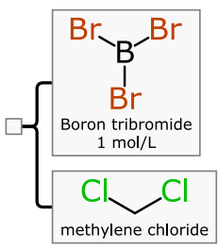
(a)




MInChI=0.00.1S/C7H8N4O2/c1-10-3-8-5-4(10)6(12)11(2)7(13)9-5/h3H,1-2H3,(H,9,13)/n1/g99pp0

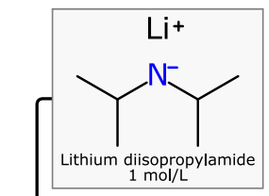
header | structure identifier | indexing | concentration

(b)

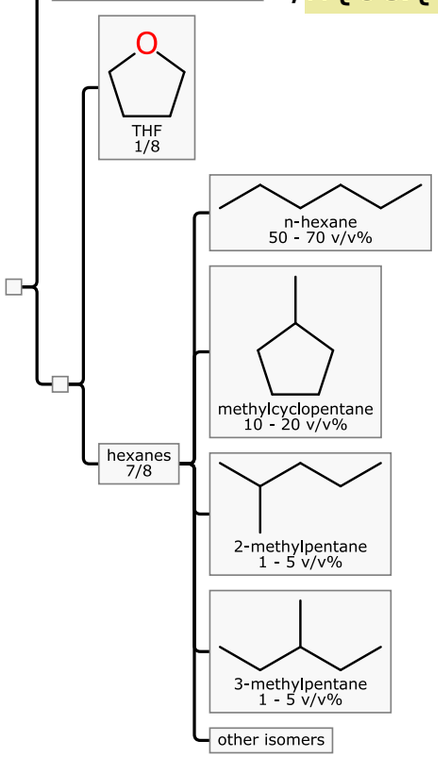



MInChI=0.00.1S/BBr3/c2-1(3)4&CH2Cl2/c2-1-3/h1H2/n{1&2}/g{1mr0&}

(c)




MInChI=0.00.1S/C4H8O/c1-2-4-5-3-1/h1-4H2&C6H12/c1-6-4-2-3-5-6/h6H,2-5H2,1H3
&C6H14/c1-3-5-6-4-2/h3-6H2,1-2H3&C6H14/c1-4-5-6(2)3/h6H,4-5H2,1-3H3
&C6H14/c1-4-6(3)5-2/h6H,4-5H2,1-3H3&C6H14N.Li/c1-5(2)7-6(3)4;/h5-6H,1-4H3;/q-1;+1
/n{6&{1&{3&2&4&5}}}/g{1mr0&{1vp0&{5:7vf-1&1:2vf-1&1:5vf-2&1:5vf-2}7vp0}}




- *Mixfile* to *MInChI*: designed for one way conversion (verbose-to-concise)
- Reduced information content of *MInChI* adds value for certain use cases
- Proof of concept implemented

# Future Work: ELNs

- Embed convenient UI within web ELN
  - sketch out mixture definitions for procedure writeup
  - quick lookup databases of known mixtures
  - search content using precise mixture-aware queries
- Vendor integration
  - work with vendors to markup their products
  - scientists can use product code to embed mixture

# Future Work: Text Extraction

- Initial proof of concept is effective but crude
- Planning a much more sophisticated iterative-learning strategy: start by formulating input for recurrent deep neural network

**Trimethyl(trifluoromethyl)silane solution 2 M in THF**

■ *component* ■ *quantity* ■ *branch* ■ *superfluous*

- Learning from input/ output data rather than curated rules: more scalable

```
 ... -5 -4 -3 -2 -1  0 +1 +2 +3 +4 +5 ...
[...  *, *, *, *, *, T, r, i, m, e, t ...] = ■ component
[...  *, *, *, *, T, r, i, m, e, t, h ...] = ■ component
[...  *, *, *, T, r, i, m, e, t, h, y ...] = ■ component
[...  *, *, T, r, i, m, e, t, h, y, l ...] = ■ component
            ...
[...  l, u, t, i, o, n,  , 2, M,  , i ...] = ■ superfluous
[...  u, t, i, o, n,  , 2, M,  , i, n ...] = ■ superfluous
[...  t, i, o, n,  , 2, M,  , i, n,  ...] = ■ quantity
[...  i, o, n,  , 2, M,  , i, n,  , T ...] = ■ quantity
          ... etc. ...
```

# Future Work: Molecules

- Structures currently limited to ≈ Molfile/InChI subset
- Want to extend to:
  - inorganics (non-integral bond orders)
  - polymers (repeat units & distributions)
  - variations (Markush structures, partially definitions)
  - large molecules (proteins, DNA)
  - pseudomolecules (ceramics, alloys)
- More formal definition of ID codes (CAS, PubChem, etc.)

# Acknowledgments


- Leah McEwen
  - and the rest of the IUPAC/InChI working group
- Hande Küçük McGinty (and iCorps)
  - Collaborative Drug Discovery

Funding
**NIH SBIR**