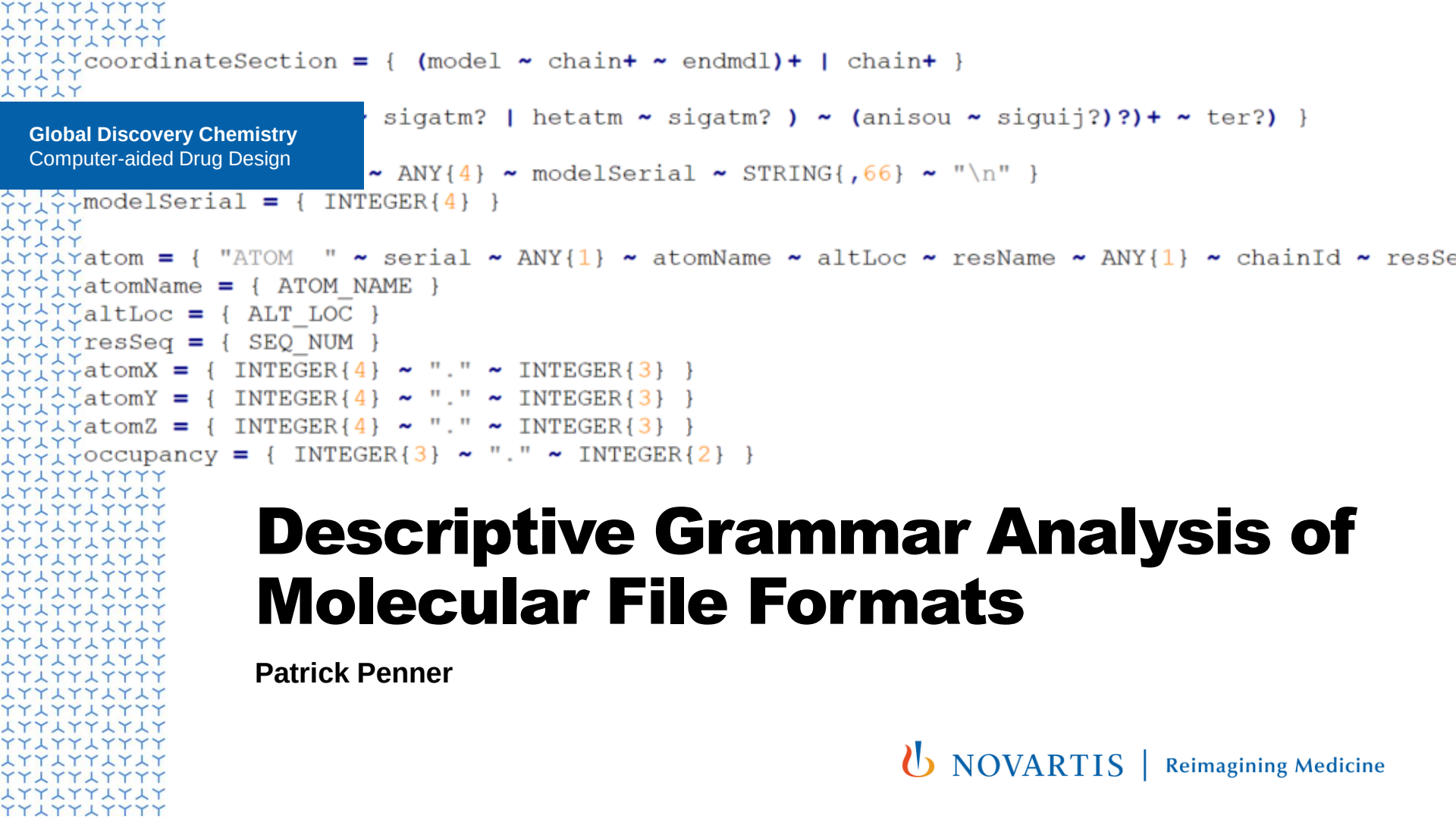


Global Discovery Chemistry
Computer-aided Drug Design

Descriptive Grammar Analysis of Molecular File Formats

Patrick Penner

NOVARTIS | Reimagining Medicine

Grammar-based vs. Hand-written Parsing

```
parity = { "@" ~ "@"? }
```

```
bond    = { ( "-" | "=" | "#" | "/" | "\\" ) }
```

```rust
fn atom_parity(scanner: &mut Scanner) -> Option<AtomParity> {
    if scanner.take(&'@') {
        if scanner.take(&'@') {
            Some(AtomParity::Clockwise)
        } else {
            Some(AtomParity::Counterclockwise)
        }
    } else {
        None
    }
}
```

```rust
pub fn bond(scanner: &mut Scanner) -> Option<BondKind> {
    scanner.transform(|target| match target {
        '-' => Some(BondKind::Single),
        '=' => Some(BondKind::Double),
        '#' => Some(BondKind::Triple),
        '/' => Some(BondKind::Up),
        '\\' => Some(BondKind::Down),
        _ => None,
    })
}
```


[1] Balsa reference implementation: https://github.com/metamolecular/balsa

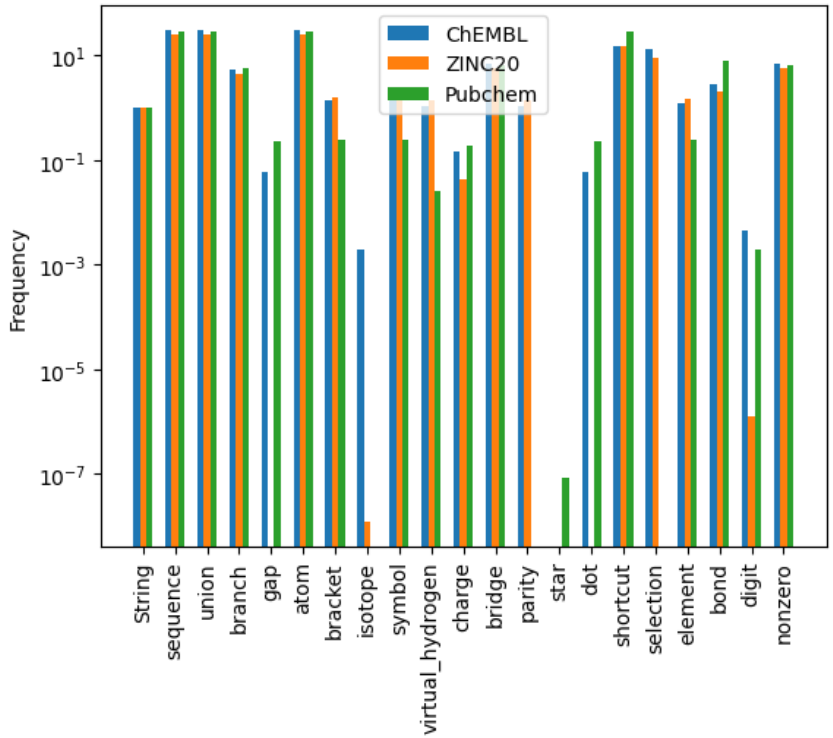
Databases

	ChEMBL	ZINC20	PubChem	PDB
Records	> 2 000 000	~ 900 000 000	> 100 000 000	> 200 000
Version	ChEMBL 33	2023-08-14	2023-08-17	2023-08-11
File type(s)	SMILES, SDF	SMILES, SDF	SMILES, SDF	PDB

 [2] Apodaca 2022

Parsing SMILES with Balsa

- Very easy to set up and largely successful
- Cannot parse ChEMBL records with: “se” and “te” (capitalization issue)
- Cannot parse PubChem records with: Md, Bh, Hs, Sg, Db, Mt
- Different domains of the databases are visible on a syntax level
  - PubChem canonical SMILES do not use stereochemistry / isotopes
  - ZINC20 does not contain salts



Parsing SD Files

```
 OpenBabel09151815293D
IIPPPPPPPPMMDDYYHHmmddSSssssssssssEEEEEEEEEEEERRRRRR
```

```
User's first and last initials (I),
program name (P),
```

Parse result:

```
{"user_initials":[" O"],"program_name":["penBabel"], ...}
```

- Meta-data is often not to specification
- Different vendors have different dialects (example: atom and bond record lengths)
- Features deprecated in the documentation may still be in use
  - Such as atom alias in next slide

 [3] Dassault Systèmes 2020

Parsing SD Files

```
135476785
  -OEChem-08222313152D

  1  0  0     0  0  0  0  0  0999 V2000
    2.0000    0.0000    0.0000 *   0  0  0  0  0  0  0  0  0  0  0  0
A    1
Cn
M  END
```

 [3] Dassault Systèmes 2020

Parsing PDB

wwPDB Format version 3.3: Introduction

If a comma, colon, or semi-colon is used in any context other than as a delimiting character, then the character must be escaped, i.e., immediately preceded by a backslash, "\".

wwPDB Format version 3.3: Title Section

MDLTYP

Example

```
 1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
MDLTYP    MINIMIZED AVERAGE
MDLTYP    CA ATOMS ONLY, CHAIN A, B, C, D, E, F, G, H, I, J, K ; P ATOMS ONLY,
MDLTYP   2 CHAIN X, Y, Z
MDLTYP    MINIMIZED AVERAGE ; CA ATOMS ONLY, CHAIN A, B
```

Examples contain un-escaped commas in a context where semi-colons are list delimiters.

Parsing PDB

Records deprecated in 1998 with no description in 2012 specification found by parsing the whole PDB.

SLTBRG

```
SLTBRG       OE1 GLU A 695                  NZ  LYS A 822   1555    1555
SLTBRG       OE1 GLU B 695                  NZ  LYS B 822   1555    1555
```

HYDBND

```
HYDBND       O3  STR A   1                   NE2 GLN A  725 1555    1555
HYDBND       O3  STR B   2                   NE2 GLN B  725 1555    1555
```

Conclusions

- Domain of the data will influence the syntax usage
- Meta-data is often not to specification
- Different vendors have different dialects
- Features deprecated in the documentation may still be in use
- Expect inconsistencies in written documentation
- Test parsers at scale

Acknowledgements

Special thanks to:

- Anna Vulpetti
- Paolo Tosco

Code: https://github.com/PatrickPenner/mol-parsing

Other work in ^{19}F fragment-based screening:

QM Assisted ML for 19F NMR Chemical Shift Prediction

https://github.com/PatrickPenner/lefshift

https://github.com/PatrickPenner/lefqm

References

1. Balsa reference implementation: https://github.com/metamolecular/balsa
2. Apodaca, R. *Balsa: A Compact Line Notation Based on SMILES*. **2022**. https://doi.org/10.26434/chemrxiv-2022-01ltp
3. Dassault Systèmes **2020** *CTFILE FORMATS* (link, date accessed: 2023-08-27)
4. wwPDB **2012** *PDB File Format - Contents Guide Version 3.30* (link, date accessed: 2023-08-27)

Grammar-based vs. Hand-written ChEMBL SMILES Benchmark

- OS: Debian GNU/Linux 11 (bullseye)
- CPU: Intel(R) Core(TM) i5-9400F CPU @ 2.90GHz
- Hard drive: 970 EVO NVMe© M.2 SSD 1TB
- ChEMBL 33: 2 231 815 SMILES

Balsa Reference Implementation		**Pest Grammar Implementation**	
real	0m17.038s	real	2m5.329s
user	0m17.038s	user	0m54.075s
sys	0m0.878s	sys	1m11.157s