chemrxn-cleaner: A Lightweight Toolkit for Parsing, Cleaning, and Preparing Organic Reaction Data for Machine Learning

2025-12-08

Summary

Machine learning models for reaction prediction, yield estimation, and synthetic planning typically require large, diverse, and standardized reaction datasets. However, widely used data sources such as the USPTO reaction corpus (Lowe 2017), the Open Reaction Database (ORD) (Schwaller et al. 2021), and custom laboratory exports (Schleinitz et al. 2022; Liu et al. 2024) vary significantly in structure, metadata availability, and quality. Before these datasets can be used for modeling or analytics, practitioners must address challenges such as inconsistent SMILES representations(Weininger 1988), missing role assignments, noisy or malformed entries, non-standard metadata formats, and heterogeneous file structures.

chemrxn-cleaner is a lightweight, extensible Python package that provides uniform tools for loading, parsing, cleaning, filtering, reporting, and exporting reaction data. It enables researchers to rapidly transform raw reaction datasets into standardized ML-ready representations with minimal boilerplate, while also supporting custom formats and user-defined domain rules. The package aims to bridge a critical but often under-documented part of the reaction-ML workflow: *data preparation*.

Statement of Need

Reaction-focused ML research depends heavily on data quality. Even minor inconsistencies in reactant–reagent separation, SMILES formatting, or metadata propagation can degrade downstream model performance. Existing cheminformatics toolkits (e.g., RDKit) provide reaction parsing primitives but do not offer a complete, opinionated pipeline for dataset-level ingestion and cleaning. Meanwhile, large datasets such as ORD or USPTO require specialized logic to interpret their schema or metadata.

chemrxn-cleaner addresses these gaps by offering:

- A **unified interface** to load heterogeneous data sources (USPTO `.rsmi`, ORD `.pb/.pb.gz`, CSV, JSON, and custom formats).
- A **standard ReactionRecord dataclass** to hold structured reaction information, conditions, yields, and arbitrary metadata.
- A suite of **filtering and cleaning utilities** to remove noisy entries, canonicalize SMILES, and encode domain heuristics.
- **Reporting tools** that summarize dataset statistics before and after cleaning.
- Optional **PyTorch dataset utilities** for rapid experimentation in reaction-prediction models.

These capabilities allow researchers to focus on modeling and analysis rather than data wrangling, which is often a major barrier to reproducibility in reaction ML projects.

State of the Field

Reaction-ML workflows in academia and industry rely heavily on data sources such as the USPTO patents dataset(Schwaller et al. 2021), curated proprietary corpora (Open Reaction Database Project 2021), or structured records from the Open Reaction Database (Kearnes et al. 2021). While several libraries provide downstream modeling components (e.g., reaction prediction architectures, graph neural networks, see (Thakkar et al. 2020; Kannas et al. 2022; Genheden et al. 2023; Wigh et al. 2024)), few open-source tools directly address the *data preparation* stage. Most publications either write custom scripts or use ad hoc pipelines that are rarely reusable across datasets.

chemrxn-cleaner complements existing chemistry toolkits (e.g., RDKit, ord-schema) by targeting the often ad-hoc "last mile" of dataset ingestion, cleaning, and transformation. It provides a registry-based loader system and a unified ReactionRecord abstraction that together support heterogeneous sources such as USPTO .rsmi, ORD .pb, CSV/JSON tables, and user-defined formats. Cleaning is expressed as a composable stack of Python callables, making it straightforward to encode domain-specific rules—for example, excluding reactions with forbidden elements, enforcing SMILES length limits, or applying similarity-based criteria. A lightweight reporting layer quantifies how each pipeline affects dataset size and composition, and minimal ML utilities (pandas conversion, deterministic splits, PyTorch datasets) make the cleaned records easy to integrate with downstream modelling libraries without prescribing any particular architecture. In this way, chemrxn-cleaner standardizes workflows that are currently re-implemented in many research codebases while remaining deliberately lightweight and library-oriented rather than a monolithic pipeline.

Software Description

Core Features

1. Unified data loading

The `load_reactions` function dispatches to format-specific loaders based on a registry. Built-in formats include:

- **USPTO `.rsmi` files**, with optional metadata preservation.
- **ORD `.pb`/`.pb.gz` files**, extracting identifiers, conditions, yields, and reaction indices.
- **CSV and JSON**, with flexible column mapping and user-provided mappers.
- **Custom formats**, registered with `register_input_format`.

Each loader produces a list of `ReactionRecord` objects.

2. Structured reaction representation

`ReactionRecord` stores:

- Parsed SMILES fields (`reaction_smiles`, `reactants`, `reagents`, `products`)
- Optional reaction conditions (temperature, time, solvents, catalysts, additives, pressure, pH, atmosphere, scale)
- Yield information
- Identifiers and provenance
- Arbitrary metadata through `extra_metadata`

Records can be serialized via `to_dict` / `from_dict` and visualized with RDKit.

3. Cleaning and filtering

The package includes:

- `clean_reactions` and `clean_and_canonicalize` for parsing and canonicalization.
- Built-in filters such as:
  - `has_product`
  - `all_molecules_valid`
  - `max_smiles_length`
  - `element_filter`
  - `meta_filter`
  - RDKit-based `similarity_filter`

Filters are simple callables returning booleans and may be freely composed.

4. Reporting and exporting

The `reporter` module exposes the `CleaningStats`/`FilterStats` structures used by the cleaning pipeline to track how many reactions were kept, dropped, or failed to parse. Cleaned reactions can be exported to JSON or CSV via `export_reaction_records`.

5. ML utilities

For rapid experimentation, the package includes:

- `records_to_dataframe` for exploratory analysis.
- `train_valid_test_split` for deterministic dataset partitioning.
- `ForwardReactionDataset`, a minimal PyTorch dataset for forward prediction tasks.

Example

```python
from chemrxn_cleaner import (
    load_reactions,
    clean_reactions_with_report,
    export_reaction_records,
)

raw = load_reactions("data/sample.rsmi", fmt="uspto", keep_meta=True)

cleaned, stats = clean_reactions_with_report(raw)
print(f"Kept {stats.n_output}/{stats.n_input} reactions after cleaning")

export_reaction_records(cleaned, "cleaned.json", fmt="json")
export_reaction_records(cleaned, "cleaned.csv", fmt="csv")
```

Acknowledgements

The package builds upon RDKit(Greg Landrum et al. 2025) for cheminformatics operations and ord-schema (Kearnes et al. 2021) for parsing ORD records. We thank contributors from the open-source chemistry community whose tools made this project possible.

References

Genheden, Samuel, Per-Ola Norrby, and Ola Engkvist. 2023. "AiZynthTrain: Robust, Reproducible, and Extensible Pipelines for Training Synthesis Prediction Models." *Journal of Chemical Information and Modeling* 63 (7): 1841–46. https://doi.org/10.1021/acs.jcim.2c01486.

Greg Landrum, Paolo Tosco, Brian Kelley, et al. 2025. *Rdkit/Rdkit: 2025_09_3 (Q3 2025) Release*. Zenodo. https://doi.org/10.5281/ZENODO.17746401.

Kannas, Christos, Amol Thakkar, Esben Bjerrum, and Samuel Genheden. 2022. *Rxnutils – a Cheminformatics Python Library for Manipulating Chemical Reaction Data*. https://doi.org/10.26434/chemrxiv-2022-wt440-v2.

Kearnes, Steven M., Michael R. Maser, Michael Wleklinski, et al. 2021. "The Open Reaction Database." *Journal of the American Chemical Society* 143 (45): 18820–26. https://doi.org/10.1021/jacs.1c09820.

Liu, Pengfei, Jun Tao, and Zhixiang Ren. 2024. *A Self-Feedback Knowledge Elicitation Approach for Chemical Reaction Predictions*. arXiv. https://doi.org/10.48550/arXiv.2404.09606.

Lowe, Daniel. 2017. *Chemical Reactions from US Patents (1976-Sep2016)*. Figshare. https://doi.org/10.6084/M9.FIGSHARE.5104873.V1.

Open Reaction Database Project. 2021. *The Open Reaction Database*. https://open-reaction-database.org.

Schleinitz, Jules, Maxime Langevin, Yanis Smail, Benjamin Wehnert, Laurence Grimaud, and Rodolphe Vuilleumier. 2022. "Machine Learning Yield Prediction from NiCOlit, a Small-Size Literature Data Set of Nickel Catalyzed c–o Couplings." *Journal of the American Chemical Society* 144 (32): 14722–30. https://doi.org/10.1021/jacs.2c05302.

Schwaller, Philippe, Alain C. Vaucher, Teodoro Laino, and Jean-Louis Reymond. 2021. "Prediction of Chemical Reaction Yields Using Deep Learning." *Machine Learning: Science and Technology* 2 (1): 015016.

Thakkar, Amol, Thierry Kogej, Jean-Louis Reymond, Ola Engkvist, and Esben Jannik Bjerrum. 2020. "Datasets and Their Influence on the Development of Computer Assisted Synthesis Planning Tools in the Pharmaceutical Domain." *Chemical Science* 11 (1): 154–68. https://doi.org/10.1039/c9sc04944d.

Weininger, David. 1988. "SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules." *Journal of Chemical Information and Computer Sciences* 28 (1): 31–36. https://doi.org/10.1021/ci00057a005.

Wigh, Daniel S., Joe Arrowsmith, Alexander Pomberger, Kobi C. Felton, and Alexei A. Lapkin. 2024. "ORDerly: Data Sets and Benchmarks for Chemical Reaction Data." *Journal of Chemical Information and Modeling* 64 (9): 3790–98. https://doi.org/10.1021/acs.jcim.4c00292.