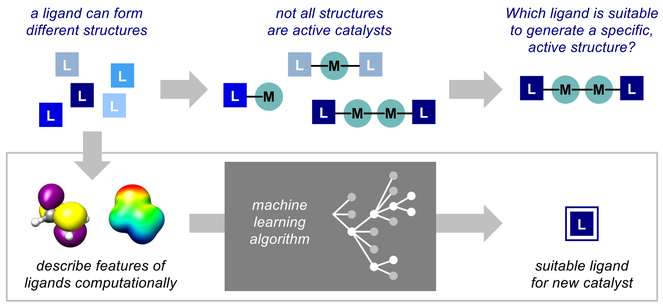
Automated Generation of Input Data for Machine-Learning-Based Predictions of Ni(I) Dimer Formation

Nina Löseke*, Sebastian Wellig+, Jannis Klinkenberg*, Christian Terboven* and Franziska Schoenebeck+
*Chair for Computer Science 12 - High Performance Computing, RWTH Aachen University
+Schoenebeck Group, Institute of Organic Chemistry, RWTH Aachen University

Motivation

- Desire to find new, reactive Ni(I) dimers for novel catalysts, but difficult to explore strategically
  - ➢ Species = groups of nickel and ligands (ions/molecules connected to metal ions) that form Ni(I) dimers
- Discern more strategic approach or properties of suitable ligands through machine learning
- ML input previously based on experiments
  - ➢ Create larger, more varied data set through DFT calculations (see below) for more innovative results



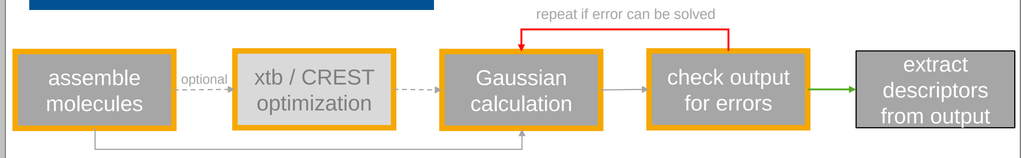
Tools

Open Babel
- Python API
- Convert chemical file formats
- Represent and assemble molecules programmatically
- Support for substructure search

xtb/CREST
- Chemical, OpenMP-parallelized software
- Local/global geometry optimization to find optimal molecular conformation with minimal total energy

Gaussian
- OpenMP-parallelized, DFT-based programs
  - ➢ DFT = “density functional theory”, which is a method for quantum mechanical modeling
- Global geometry optimization
- Calculations for molecular descriptors that are used as ML features for input data

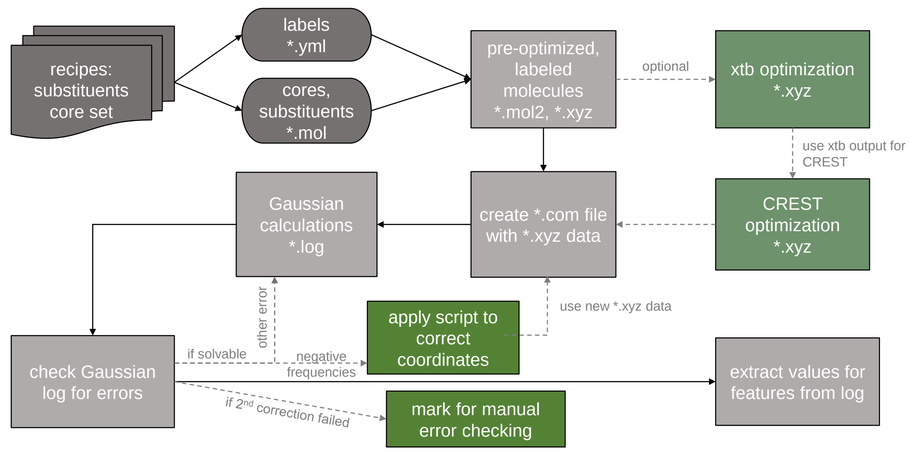
Workflow and Status Quo



Status Quo
- Previous, partial implementation by Schoenebeck Group only for orange part
- A lot of manual interaction and monitoring of batch jobs needed
  - ➢ Adapt templates for xtb/CREST/Gaussian input and batch jobs manually
  - ➢ Set input paths to assemble each molecule from cores and substituents
  - ➢ Check Gaussian logs for errors manually

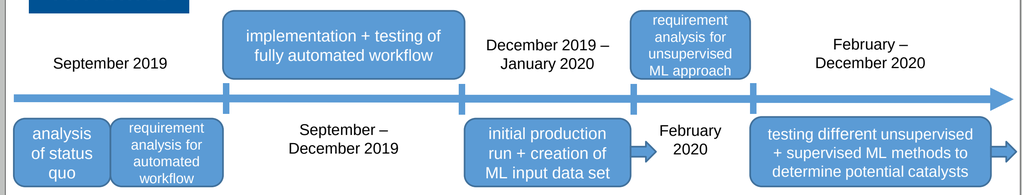
Possible Improvements for Fully Automated Framework
- Parallelize work for several structures
- Handle implicit dependencies between steps automatically
- Define general recipe format for non-chemist users to create structure library from ligand library
- Avoid hard-coded paths in scripts; fill in placeholders in templates automatically

Fully Automated Framework



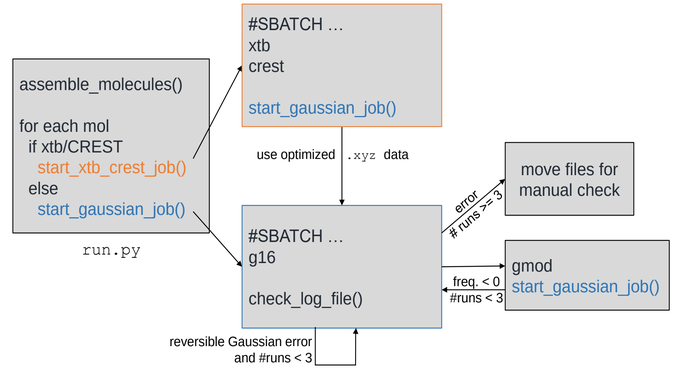
- Python-based framework spread across several directories; input provided by user
- Run all steps at once or each workflow step individually; feature extraction is run independently
- Steps run in parallel across structure library
- Logging, marking errors; automated clean-up after each successful Gaussian run
- Final output: `pandas.DataFrame` fit for ML algorithms

Timeline



Batch Job Dependencies

- Run next batch job/script from within previous batch job/script
  - ➢ Manage implicit dependencies



Conclusion and Outlook

Conclusion
- Increased efficiency and less error-prone due to automation
- Automated workflow eliminates 90% of manual effort per molecule
- Automatic handling of dependencies
- Flexible and customizable DFT part
- Parallel use of resources (OpenMP-parallelized batch jobs)
- Output compatible with many ML algorithms

Outlook
- On-going work on project: currently testing different ML models to evaluate generated data; integration of new molecular data into ligand and recipe library; continuous adaption of framework to match particularities of HPC cluster
- Offer tool to computational chemists and other scientists who also use Gaussian programs for simulations
- Workflow design including batch-job creation and management can be adapted by other HPC users, especially for AI/ML-related HPC applications regarding generation of input
- Enable use of ML in field of chemistry by means of larger datasets

Contact

Nina Löseke
Chair for Computer Science 12 - HPC, RWTH Aachen
loeseke@itc.rwth-aachen.de
Sebastian Wellig
Schoenebeck Group, IOC, RWTH Aachen
sebastian.wellig@rwth-aachen.de

Part of the Python scripts, the input data and graphic for “Motivation” were kindly provided by the Schoenebeck Group.