Icolos: A workflow manager for structure based post-processing of de novo generated small molecules

J. Harry Moore[#], Matthias R. Bauer[§], Jeff Guo[#], Atanas Patronov[#], Ola Engkvist[#,$] and Christian Margreitter[#,*]

Summary

We present Icolos, a workflow manager written in Python as a tool for automating complex structure-based workflows. Icolos can be used as a standalone tool, for example in virtual screening campaigns, or can be used in conjunction with deep learning-based molecular generation facilitated for example by REINVENT, a previously published *de novo* design package. In this publication, we focus on the internal structure and general capabilities of Icolos, using docking experiments as an illustration.

Availability and Implementation

The source code is freely available at https://github.com/MolecularAI/Icolos under the Apache 2.0 licence. Tutorial notebooks containing minimal working examples can be found at https://github.com/MolecularAI/IcolosCommunity.

Contact

*christian.margreitter@astrazeneca.com

Supplementary information

A detailed description of the package, including common use cases, is provided in the SI.

Affiliations

… Molecular AI, Discovery Sciences, R&D, AstraZeneca, Gothenburg, Sweden
§ … Structure & Biophysics, Discovery Sciences, R&D, AstraZeneca, Cambridge, UK
$ … Department of Computer Science and Engineering, Chalmers University of Technology, Gothenburg, Sweden

Introduction

Structure-based computational methods provide significant predictive insight and see widespread use from hit discovery to lead optimisation. However, manual execution of multistep workflows is inefficient, labour intensive and error prone, especially when stitching together multiple programs *ad hoc* using scripting languages such as bash.

Existing workflow managers, such as Biovia's PipelinePilot or KNIME, are commonly used to automate such tasks in a more standardized way.[1,2] However, they have certain limitations that led us to develop a new solution, specifically, we required a tool that would seamlessly integrate with our deep learning based molecular *de novo* design tool, REINVENT, to construct complex scoring components through vendor agnostic workflows while being flexible enough to support rapid prototyping.[3]

Here we present Icolos, a modular, flexible and extensible workflow manager that provides a unified interface to a host of common commercial and open-source computational packages, encompassing docking, molecular dynamics, binding free energy and quantum mechanical calculations. Icolos has built-in REINVENT integration and has been used in-house both to incorporate complex structural calculations into the agent's feedback loop, and as a standalone workflow manager for post-processing results. We achieve efficient scaling through parallelisation of

computationally demanding calculations and performance in agreement with manually executed workflows, often at a fraction of the runtime of previous implementations relying on shell scripts or submission from a GUI.

Software Implementation

Icolos workflows are constructed as a list of elementary "steps" which defines the flow of information through the workflow. Over 40 individual steps are currently supported, covering a wide variety of commercial and open-source software, which can be combined in arbitrary order. Templates for common workflows are available and can be readily extended or adapted. In principle, any program that provides a command-line executable or Python API could be incorporated as a workflow step.

Workflow configurations are specified in a JSON file, with each step defined by a standardised set of fields, controlling the execution environment, parallelization scheme, error handling and settings to control both the underlying program and the step's execution. Typically, all underlying command-line options of the backends are accessible which allows a high degree of control.

Since many workflows implement virtual screening capabilities on libraries of compounds, Icolos efficiently handles molecules in a multi-tier data structure comprised of compounds, enumerations and conformers. This is based on the RDKit Mol object, and keeps track of computed properties as the workflow progresses.[4]

This provides the basis for flexible write-out capabilities in a variety of standard formats and allows for efficient parallelization at both the step and workflow level to leverage high-performance computing resources.

In general, steps that perform computations on a set of compounds can be parallelized across multiple cores, and tasks can be either run directly utilizing the master job's resources, or on a SLURM cluster through the integrated submission and monitoring interface. This enables execution of heterogeneous workflows requiring both CPU and GPU resources (for example a combined docking and molecular dynamics workflow), with efficient use of cluster resources.

In this work, we introduce an ensemble docking workflow using Icolos, in which ligands are docked against multiple receptor grids, which can be constructed from either different crystal structures or central member structures obtained from a molecular dynamics trajectory. In our experience, ensemble docking can lead to substantial ligand enrichment compared to a single grid, especially where there is significant receptor flexibility. The full workflow consists of the following steps, and is summarised in Figure 1:

- Ligand embedding, enumerating possible protonation states, stereo-chemistry and tautomers,
- Docking against multiple grids
- Filtering and reporting back the best score per compound across all receptor structures.

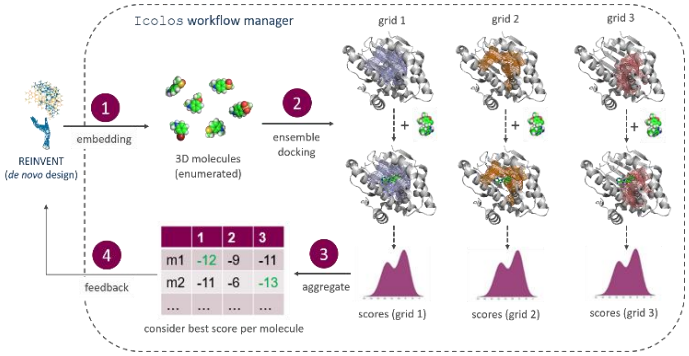

Figure 1: Graphical summary of ensemble docking workflow implemented in Icolos

An example step configuration for ensemble docking with AutoDock Vina is shown below.[5] For details and the full workflow configuration files, including the use of alternative ligand preparation and docking backends, we refer to the SI and IcolosCommunity repository.

```
{
  "step_id": "adv_docking_example",
  "type": "vina_docking",
  "execution": {
    "prefix_execution": "module load AutoDock_Vina",
    "parallelization": {
         "cores": 4
         },
         "failure_policy": {
           "n_tries": 3
         }
  },
  "settings": {
    "arguments": {
         "flags": [],
         "parameters": {
         }
    },
    "additional": {
         "configuration": {
           "receptor_path ": [
             "<path/to/grid1.zip>",
             "<path/to/grid2.zip>"
           ]
         "number_poses ": 2,
         "search_space"{
           <fill grid params>
         }
     }
  },
  "input": {
    "compounds": [{
         "source": "Ligprep",
         "source_type": "step"
    }]
  },
  "writeout": [{
```

```
    "compounds": {
        "category": "conformers"
        },
        "destination": {
          "resource": "<path/to/conformers.sdf>",
          "type": "file",
          "format": "SDF"
        }
   }]
}
```

Notably, the parallelization block divides the compounds across four cores, with one compound allocated per core per batch. Any specified flags and parameters are passed directly to Vina. The resulting conformers are written to an SDF file, with the computed docking scores annotated.

Conclusions

We have developed Icolos, a general-purpose workflow manager for structure-based workflows. Icolos has been successfully deployed internally to develop, reproduce and distribute complex workflows in drug discovery projects, and handle complex scoring components for *de novo* molecular generation using REINVENT.

More complex use cases will be benchmarked and described in detail in subsequent publications.

Acknowledgements

The authors thank Maxime Tarrago, Jon Paul Janet, Martin Packer, Luca Carlino, Magdalena Weber, Linnea Johansson and the AstraZeneca Scientific Computing Platform team for their contributions.

References

1. BIOVIA-Dassault Systèmes. PipelinePilot. (2022).
2. Berthold, M. R. *et al.* KNIME - the Konstanz information miner. *ACM SIGKDD Explor. Newsl.* 58–61 (2009) doi:10.1145/1656274.1656280.
3. Blaschke, T. *et al.* REINVENT 2.0: An AI Tool for De Novo Drug Design. (2020) doi:10.1021/acs.jcim.0c00915.
4. Landrum, G. *et al.* rdkit/rdkit: 2021_09_4 (Q3 2021) Release. (2022) doi:10.5281/ZENODO.5835217.
5. Eberhardt, J., Santos-Martins, D., Tillack, A. F. & Forli, S. AutoDock Vina 1.2.0: New Docking Methods, Expanded Force Field, and Python Bindings. *J. Chem. Inf. Model.* **61**, 3891–3898 (2021).