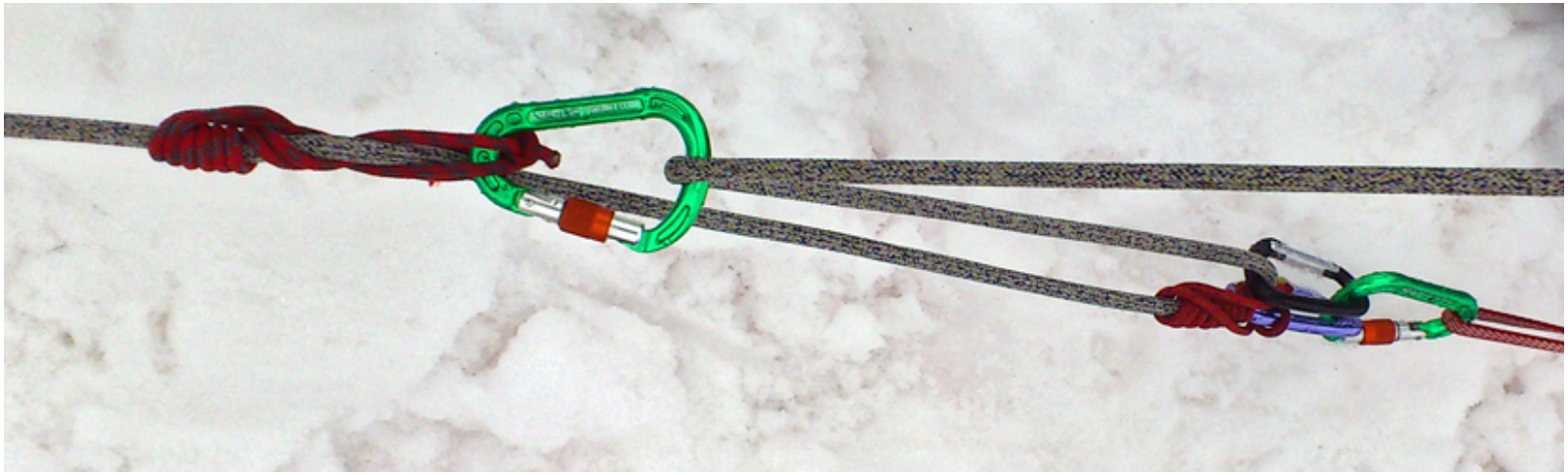


RDKit: The State of the Toolkit

Gregory Landrum

NIBR Informatics

Novartis Institutes for BioMedical Research

RDKit UGM 2014, Darmstadt

Open-Source Cheminformatics and Machine Learning

NOVARTIS

Overview

- **History**
- Where are we today?
- What next?

NOVARTIS

Presentations, tutorials, notes, data, etc.

https://github.com/rdkit/UGM_2014

Please do pull requests or just send me materials

NOVARTIS

History and milestones

- 2000-2006: initial development work at Rational Discovery
- 2006: code open sourced and released on sourceforge.net
- 2007: First NIBR contribution (chemical reaction handling); Noel discovers the RDKit (=first rdkit-discuss post?)
- 2008: first POC of Java wrapper; Mac support added; SLN and Mol2 parsers;
- 2009: Morgan fingerprints; switch to cmake; switch to VF2 for SSS
- 2010: PostgreSQL cartridge; First iteration of the KNIME nodes; $RDBASE/Contrib appears; SaltRemover and FunctionalGroups code
- 2011: New Java wrappers; more functionality moved to C++; InChI support; Avalontools integration
- 2012: First UGM; Speed improvements; MCS implementation; IPython integration; “RDKit Cookbook” appears
- 2013: Move to github; Pandas integration; MMFF and Open3DAlign support; PDB support; rdkit blog started
- 2014: python3 support; conda integration; experimental lucene integration;

Overview

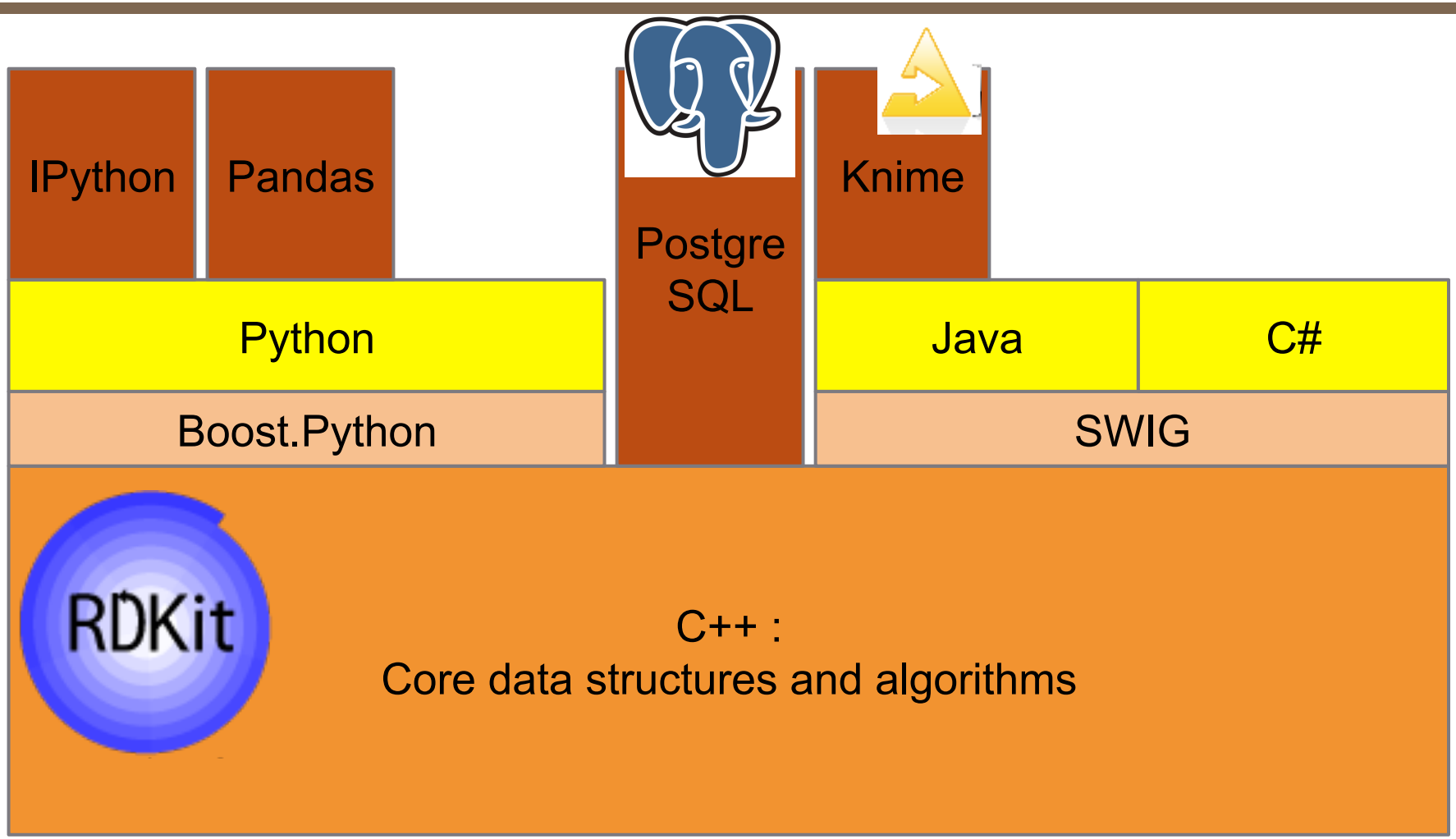
- History
- **Where are we today?**
- What next?

NOVARTIS

RDKit: What is it?

- Open-source C++ toolkit for cheminformatics
- Wrappers for Python (2.x), Java, C#
- Functionality:
  - 2D and 3D molecular operations
  - Descriptor generation for machine learning
  - PostgreSQL database cartridge for substructure and similarity searching
  - Knime nodes
  - IPython integration
  - Lucene integration (experimental)
  - Supports Mac/Windows/Linux
- Releases every 6 months
- business-friendly BSD license
- Code: https://github.com/rdkit
- http://www.rdkit.org

The RDKit “ecosystem”



Exact same algorithms/implementations accessible from many different endpoints

NOVARTIS

Some features

- Input/Output: SMILES/SMARTS, SDF, TDT, PDB, SLN [1], Corina mol2 [1]
- "Cheminformatics":
  - Substructure searching
  - Canonical SMILES
  - Chirality support (i.e. R/S or E/Z labeling)
  - Chemical transformations (e.g. remove matching substructures)
  - Chemical reactions
- 2D depiction, including constrained depiction
- 2D->3D conversion/conformational analysis via distance geometry
- UFF and MMFF94 implementation for cleaning up structures
- Fingerprinting: Daylight-like, atom pairs, topological torsions, Morgan algorithm, "MACCS keys", etc.
- Similarity/diversity picking
- 2D pharmacophores [1]
- Gasteiger-Marsili charges
- Hierarchical subgraph/fragment analysis
- Bemis and Murcko scaffold determination
- RECAP and BRICS implementations
- Multi-molecule maximum common substructure
- Feature maps
- Shape-based similarity
- Fraggle similarity (from GSK)
- Molecule-molecule alignment
- Open3DAlign implementation
- Integration with PyMOL for 3D visualization
- Functional group filtering
- Salt stripping
- Molecular descriptor library:

  Topological (κ3, Balaban J, etc.), Compositional (Number of Rings, Number of Aromatic Heterocycles, etc.), EState, SlogP/SMR (Wildman and Crippen approach), "MOE like" VSA descriptors, Feature-map vectors
- Machine Learning:
  - Clustering (hierarchical)
  - Information theory (Shannon entropy, information gain, etc.)
- Tight integration with the IPython notebook and pandas
- Integration with the InChI library

[1] These implementations are functional but are not necessarily the best, fastest, or most complete.

The contrib dir

- LEF (Anna Vulpetti, NIBR): Local Environment of Fluorine
- PBF (Nicholas Firth, ICR): Plane of best fit descriptor
- SA_Score (Peter Ertl, NIBR): synthetic-accessibility score
- fraggle (Jameed Hussain, GSK): fragment-based similarity
- mmpa (Jameed Hussain, GSK): molecular matched pairs
- pzc (Paul Czodrowski, Merck KGaA): tools for building and validating classifiers
- ConformerParser (Sereina Riniker, ETH): parser for Amber trajectory files

RDKit: Documentation?

The documentation:

The RDKit 2013.06.1 documentation » next | index

The RDKit Documentation

- An overview of the RDKit
  - What is it?
  - Functionality overview
  - The Contrib Directory
  - License
- Installation
  - Linux and the Mac
    - Installation from repositories
      - Ubuntu 12.04 and later
      - Fedora, CentOS, and RHEL
      - MacOS
    - Building from Source
      - Prerequisites
      - Building the RDKit
      - Testing the build (optional, but recommended)
      - Advanced
      - Frequently Encountered Problems
  - Windows

RDKit

Open-Source Cheminformatics and Machine Learning

Table Of Contents

Next topic

Built using Python's standard docs tool: Sphinx

NOVARTIS

RDKit: Documentation?

Sample section from introductory docs:

Reading and Writing Molecules

Reading single molecules

The majority of the basic molecular functionality is found in module rdkit.Chem:

```
>>> from rdkit import Chem
```

Individual molecules can be constructed using a variety of approaches:

```
>>> m = Chem.MolFromSmiles('Cc1ccccc1')
>>> m = Chem.MolFromMolFile('data/input.mol')
>>> stringWithMolData=file('data/input.mol','r').read()
>>> m = Chem.MolFromMolBlock(stringWithMolData)
```

All of these functions return a Mol object on success:

```
>>> m
<rdkit.Chem.rdchem.Mol object at 0x...>
```


Note: docs that include python code snippets are *tested*.

NOVARTIS

RDKit: Who is using it?

- Hard to say with any certainty
- >900 downloads of last release
- Active contributors to the mailing list from:
  - Big pharma
  - Small pharma/biotech
  - Software/Services
  - Academia
- Contributions coming from the community (wiki pages, code patches, changes to the build system, new features etc.) as well as active use in other systems.
- Community contributions for packaging:
  - rpms/debs for Fedora/Debian linux
  - homebrew recipe for MacOS
  - conda packages

NOVARTIS

Sustainability of the RDKit

… thinking about the bus problem

- This clearly isn't just a hobby project any more
- Used internally in NIBR in multiple production systems
- Contributions coming in from the community
- I'm no longer the only one answering questions on the mailing list

NOVARTIS

Overview

- History
- Where are we today?
- **What next?**

NOVARTIS

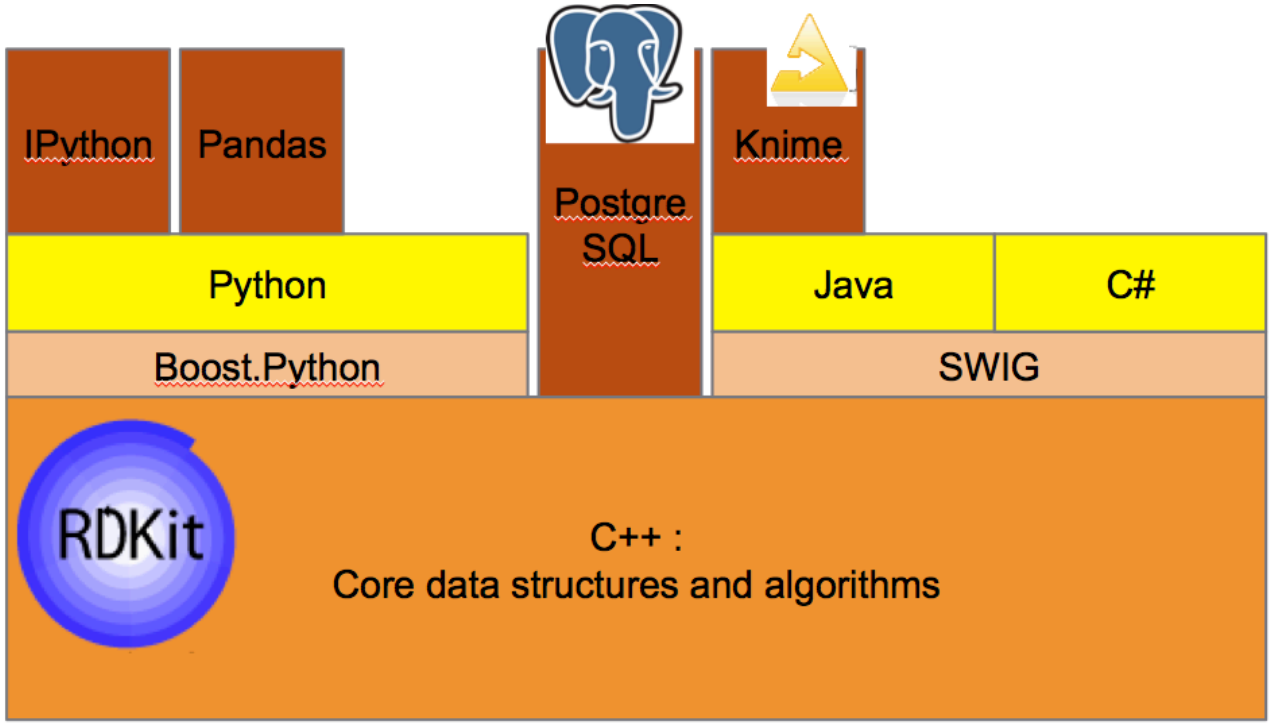
What's next?

- We'll decide some of that here in the round-table session, discussions, and the hackathon
- Obvious candidates:
  - Further performance improvements
  - Improved documentation
  - Move more code into C++ (allows access from Knime and the cartridge)
  - Better packaging
- Some things that are queued up:
  - Improved canonicalization performance
  - Molecular interaction field (MIF) implementation
  - A new molecular hash function
  - 3D shape-based alignment

Presentations, tutorials, notes, data, etc.

https://github.com/rdkit/UGM_2014

Please do pull requests or just send me materials



NOVARTIS