RDKit: The State of the Toolkit

Gregory Landrum

NIBR IT

Novartis Institutes for BioMedical Research

RDKit UGM 2013, Hinxton

Open-Source Cheminformatics and Machine Learning

NOVARTIS

Acknowledgements

- NIBR
  - Tom Digby (Legal)
  - Richard Lewis (GDC)
  - Remy Evard (NIBR IT)
  - Andy Palmer (NIBR IT)
  - Nik Stiefl (GDC)
  - Peter Gedeck (GDC)
  - Manuel Schwarze (NIBR IT)
  - Eddie Cao (NIBR IT)
  - Nikolas Fechner (NIBR IT)
  - Sereina Riniker (NIBR IT)
- The RDKit Open-Source Community
  - Andrew Dalke
  - Noel O’Boyle
  - Roger Sayle
  - Paolo Tosco
- knime.com
  - Michael Berthold
  - Bernd Wiswedel
  - Thorsten Meinl
- PostgreSQL cartridge:
  - Michael Stonebraker
  - Oleg Bartunov
  - Teodor Sigaev
  - Pavel Velikhov
- Distributions:
  - Gianluca Sforna (Fedora)
  - Michael Banck (Debian)

Overview

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

Presentations, tutorials, notes, data, etc.

https://github.com/rdkit/UGM_2013

Please do pull requests or just send me materials

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 support;

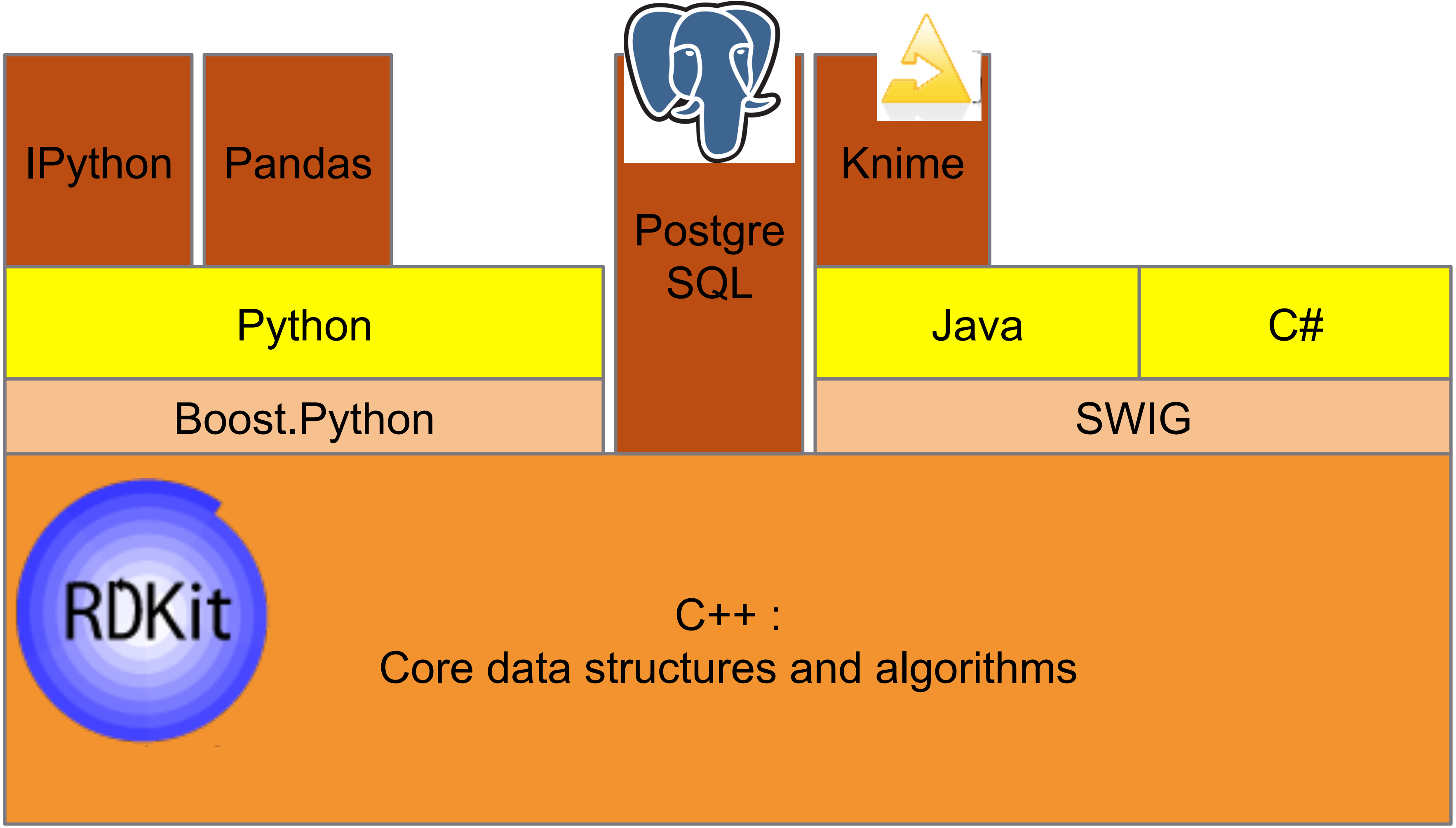
Overview

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

RDKit: What is it?

- Python (2.x), Java, and C++ toolkit for cheminformatics
  - Core data structures and algorithms in C++
  - Python wrapper generated using Boost.Python
  - Java wrapper generated with SWIG
- Functionality:
  - 2D and 3D molecular operations
  - Descriptor generation for machine learning
  - Molecular database cartridge
  - Supports Mac/Windows/Linux
- History:
  - 2000-2006: Developed and used at Rational Discovery for building predictive models for ADME, Tox, biological activity
  - June 2006: Open-source (BSD license) release of software, Rational Discovery shuts down
  - to present: Open-source development continues, use within Novartis, contributions from Novartis back to open-source version

The RDKit “ecosystem”



IPython

Pandas

Postgre SQL

Knime

Python

Java

C#

Boost.Python

SWIG

C++ :
Core data structures and algorithms

Exact same algorithms/implementations accessible from many different endpoints

What can you do with it?

A laundry list

- Input/Output: SMILES/SMARTS, SDF, TDT, 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
  - Molecular serialization (e.g. mol <-> text)
- 2D depiction, including constrained depiction
- 2D->3D conversion/conformational analysis via distance geometry
- UFF/MMFF implementations for cleaning up structures
- Fingerprinting:

  Daylight-like, atom pairs, topological torsions, Morgan algorithm, “MACCS keys”, etc.
- Similarity/diversity picking (including fuzzy similarity)
- 2D pharmacophores[1]
- Gasteiger-Marsili charges
- Hierarchical subgraph/fragment analysis
- RECAP and BRICS implementations
- Multi-molecule maximum common substructure[2]

[1] *functional, but not great implementations*

[2] *Contribution from A. Dalke*

What can you do with it?

A laundry list, cntd

- Feature maps
- Shape-based similarity
- Molecule-molecule alignment
- Shape-based alignment (subshape alignment) [1]
- Integration with PyMOL for 3D visualization
- Database integration
- Molecular descriptor library:
  - Topological (κ3, Balaban J, etc.)
  - Electrotopological state (Estate)
  - clogP, MR (Wildman and Crippen approach)
  - “MOE like” VSA descriptors
  - Feature-map vectors
- Machine Learning:
  - Clustering (hierarchical)
  - Information theory (Shannon entropy, information gain, etc.)
  - Decision trees, *naïve Bayes*[1], *kNN*[1]
  - Bagging, random forests
  - Infrastructure (data splitting, shuffling, enrichment plots, serializable models, etc.)
  - Support for standard python packages: pandas, scikit-learn

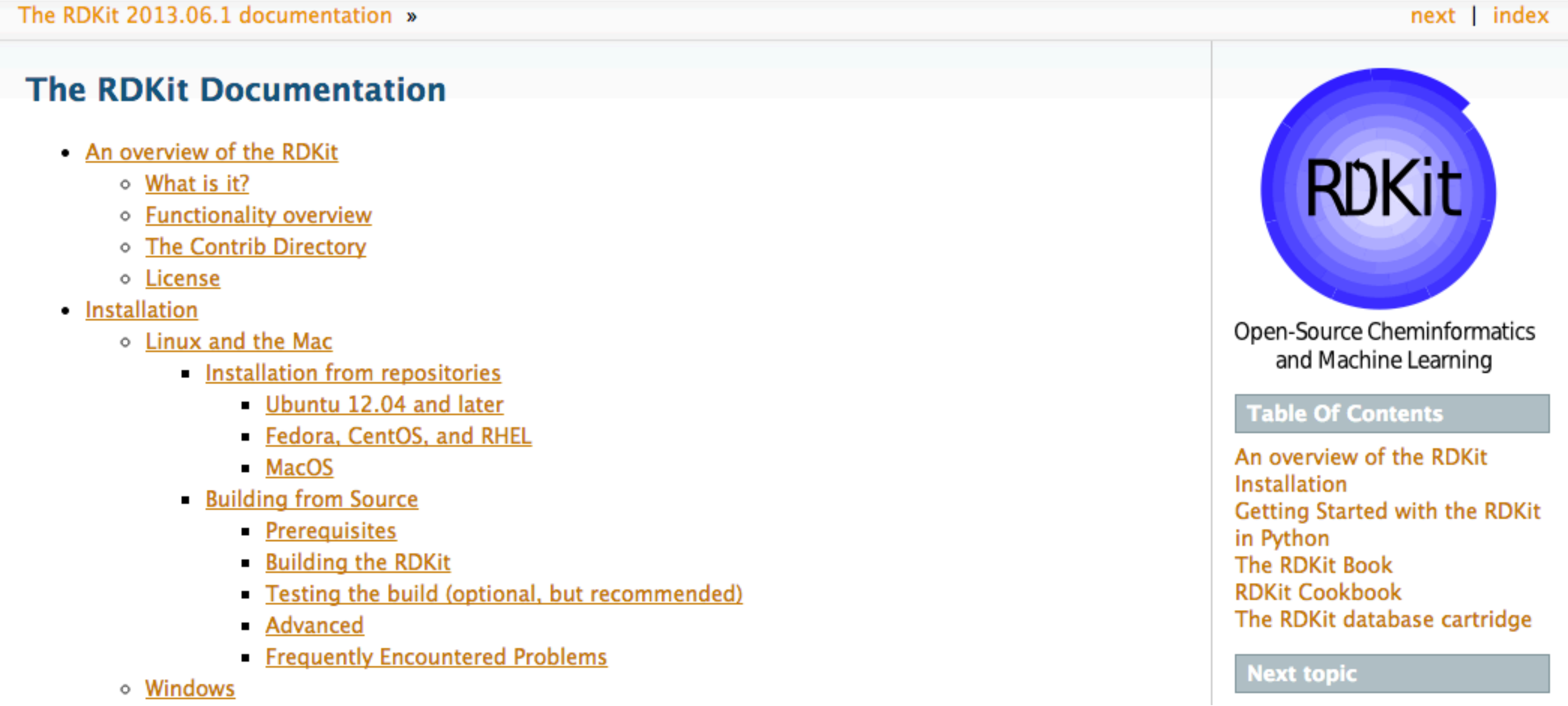
[1] *functional, but not great implementations*

RDKit: Where is it?

- Web page: http://www.rdkit.org
- github: code repository, bug tracker, downloads
  - https://github.com/rdkit/rdkit
- sourceforge: mailing lists, downloads
  - http://sourceforge.net/projects/rdkit
- Google code: wiki, downloads
  - http://code.google.com/p/rdkit/
- Releases: quarterly
- Licensing: new BSD
- Documentation:
  - HTML/PDF “Getting Started” documentation
  - in-code docs extracted by either doxygen (C++) or epydoc (python)
- Getting help:
  - Check the wiki and “Getting Started” document
  - The rdkit-discuss mailing list

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

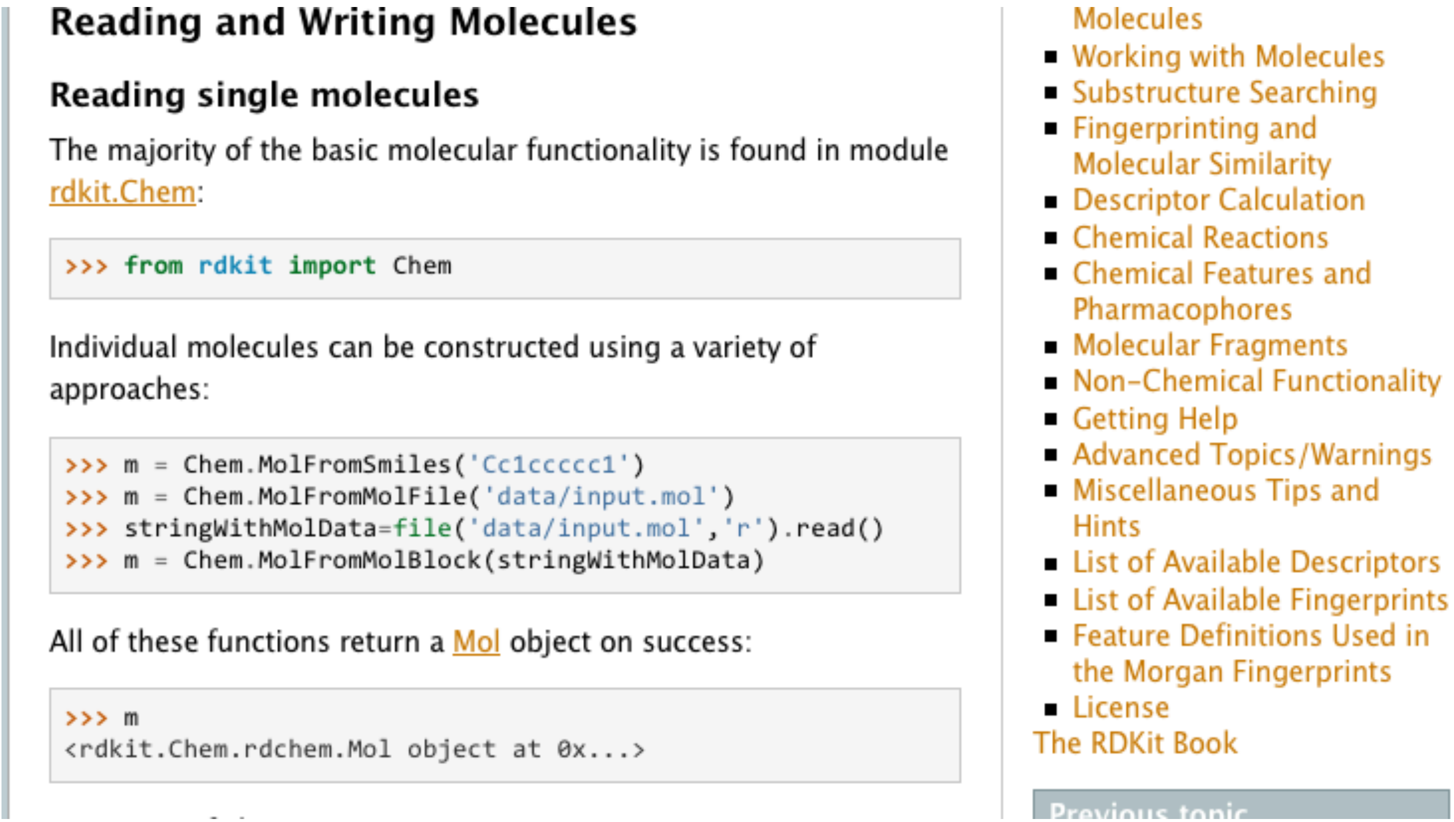
An overview of the RDKit
Installation
Getting Started with the RDKit in Python
The RDKit Book
RDKit Cookbook
The RDKit database cartridge

Next topic

Built using Python’s standard docs tool: Sphinx

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...>
```

Molecules

- Working with Molecules
- Substructure Searching
- Fingerprinting and Molecular Similarity
- Descriptor Calculation
- Chemical Reactions
- Chemical Features and Pharmacophores
- Molecular Fragments
- Non-Chemical Functionality
- Getting Help
- Advanced Topics/Warnings
- Miscellaneous Tips and Hints
- List of Available Descriptors
- List of Available Fingerprints
- Feature Definitions Used in the Morgan Fingerprints
- License

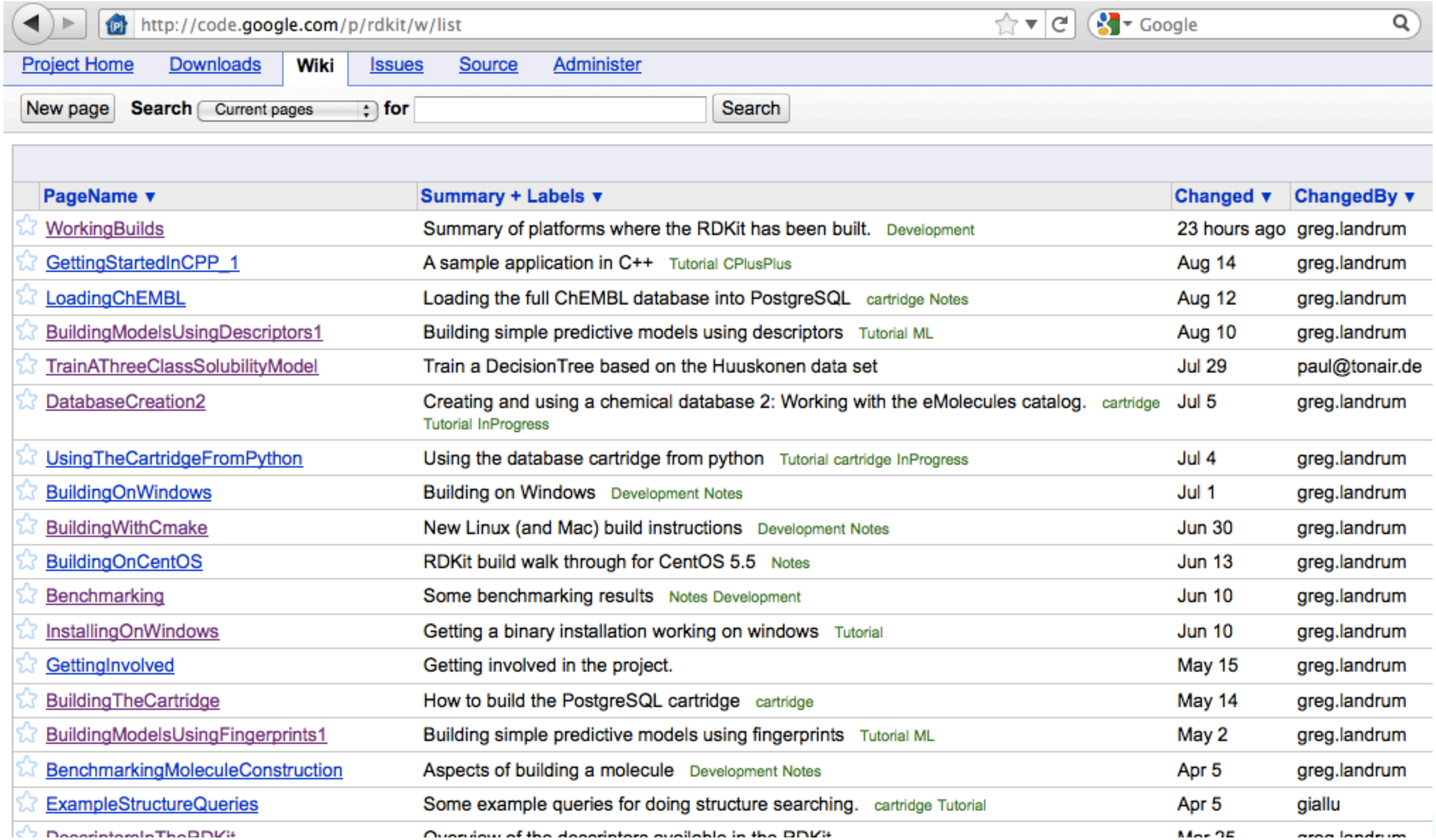
The RDKit Book

Previous topic

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

RDKit: Documentation?

The wiki: http://code.google.com/p/rdkit/w/list



PageName ▼	Summary + Labels ▼	Changed ▼	ChangedBy ▼
WorkingBuilds	Summary of platforms where the RDKit has been built. Development	23 hours ago	greg.landrum
GettingStartedInCPP_1	A sample application in C++ Tutorial CPlusPlus	Aug 14	greg.landrum
LoadingChEMBL	Loading the full ChEMBL database into PostgreSQL cartridge Notes	Aug 12	greg.landrum
BuildingModelsUsingDescriptors1	Building simple predictive models using descriptors Tutorial ML	Aug 10	greg.landrum
TrainAThreeClassSolubilityModel	Train a DecisionTree based on the Huuskonen data set	Jul 29	paul@tonair.de
DatabaseCreation2	Creating and using a chemical database 2: Working with the eMolecules catalog. cartridge Tutorial InProgress	Jul 5	greg.landrum
UsingTheCartridgeFromPython	Using the database cartridge from python Tutorial cartridge InProgress	Jul 4	greg.landrum
BuildingOnWindows	Building on Windows Development Notes	Jul 1	greg.landrum
BuildingWithCmake	New Linux (and Mac) build instructions Development Notes	Jun 30	greg.landrum
BuildingOnCentOS	RDKit build walk through for CentOS 5.5 Notes	Jun 13	greg.landrum
Benchmarking	Some benchmarking results Notes Development	Jun 10	greg.landrum
InstallingOnWindows	Getting a binary installation working on windows Tutorial	Jun 10	greg.landrum
GettingInvolved	Getting involved in the project.	May 15	greg.landrum
BuildingTheCartridge	How to build the PostgreSQL cartridge cartridge	May 14	greg.landrum
BuildingModelsUsingFingerprints1	Building simple predictive models using fingerprints Tutorial ML	May 2	greg.landrum
BenchmarkingMoleculeConstruction	Aspects of building a molecule Development Notes	Apr 5	greg.landrum
ExampleStructureQueries	Some example queries for doing structure searching. cartridge Tutorial	Apr 5	giallu
DescriptorsInTheRDKit	Overview of the descriptors available in the RDKit	Mar 25	greg.landrum

RDKit: Documentation?

The documents that come out of the tutorials here are going to be another good source of information

RDKit: Who is using it?

- Hard to say with any certainty
- ~500 downloads of each new version
- Active contributors to the mailing list from:
  - Big pharma
  - Small pharma/biotech
  - Software/Services
  - Academia
- Starting to see contributions coming from the community (wiki pages, code patches, changes to the build system, etc.) as well as active use in other systems.
- Community contributions for packaging:
  - rpms/debs for Fedora/Debian linux
  - homebrew recipe for MacOS

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 outside
- I’m no longer the only one answering questions on the mailing list

Switch to notebook

Overview

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

What’s next?

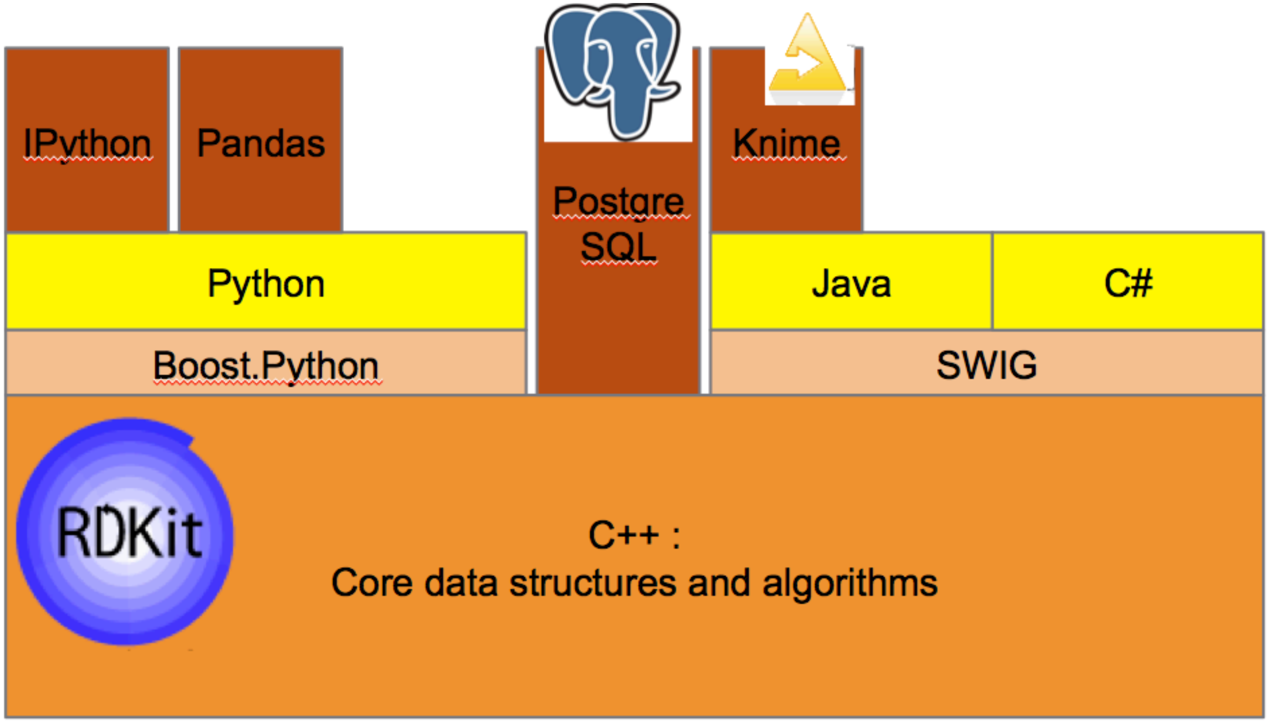
- We’ll decide some of that here in the round-table session and discussions

- Obvious candidates:
  - Further performance improvements
  - Improved documentation
  - Move more code into C++ (allows access from Knime and the cartridge)
  - Better packaging

Presentations, tutorials, notes, data, etc.

https://github.com/rdkit/UGM_2013

Please do pull requests or just send me materials


IPython
Pandas
PostgreSQL
Knime
Python
Java
C#
Boost.Python
SWIG
RDKit
C++ :
Core data structures and algorithms