Structural Bioinformatics Training Workshop & Hackathon 2017

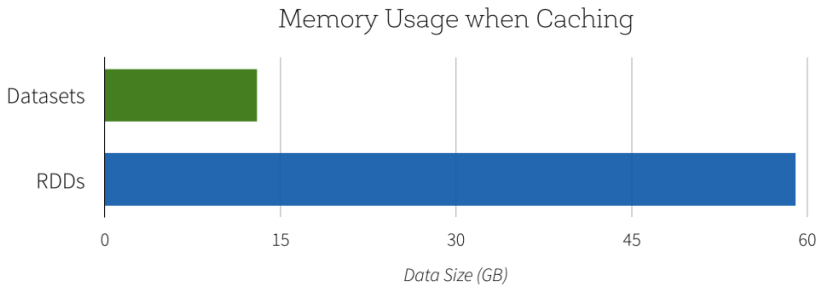
Advanced MMTF-Spark

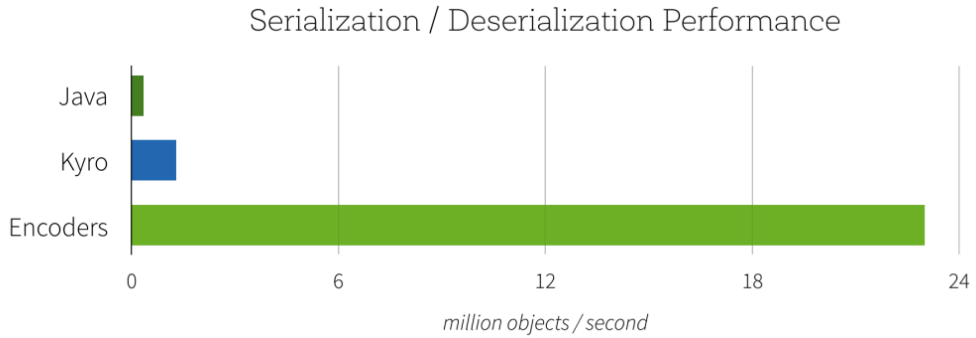
Peter Rose

Structural Bioinformatics Laboratory
San Diego Supercomputer Center
UC San Diego

Dataset

- Table of typed objects with a relational schema
- Similar to Python Pandas and R Dataframes
- Distributed data structure optimized for performance
- Distributed SQL queries on Dataset (Spark SQL)





Source: https://databricks.com/blog/2016/01/04/introducing-apache-spark-datasets.html

Custom Report of PDB Annotations

```
// spark setup
JavaSparkContext sc = …

// retrieve PDB annotation: Binding affinities (Ki, Kd),
// group name of the ligand (hetId), and the
// Enzyme Classification number (ecNo)
Dataset<Row> ds = CustomReportService.getDataset("Ki","Kd","hetId","ecNo");

// show the schema of this dataset
ds.printSchema();

// select structures that either have a Ki or Kd value(s) and
// are protein-serine/threonine kinases (EC 2.7.1.*)
// by using dataset operations
ds = ds.filter("(Ki IS NOT NULL OR Kd IS NOT NULL) AND ecNo LIKE '2.7.11.%'");
ds.show(10);
```

List of custom report fields: http://www.rcsb.org/pdb/results/reportField.do

Creating a Temporary Table/SQL

```
// spark setup
JavaSparkContext sc = …

// retrieve PDB annotation: Binding affinities (Ki, Kd),
// group name of the ligand (hetId), and the
// Enzyme Classification number (ecNo)
Dataset<Row> ds = CustomReportService.getDataset("Ki","Kd","hetId","ecNo");

// select structures that either have a Ki or Kd value(s) and
// are protein-serine/threonine kinases (EC 2.7.1.*)
// by creating a temporary query and running SQL
ds.createOrReplaceTempView("table");
ds.sparkSession().sql("SELECT * from table WHERE
(Ki IS NOT NULL OR Kd IS NOT NULL) AND ecNo LIKE '2.7.11.%'");

ds.show(10);
```

List of custom report fields: http://www.rcsb.org/pdb/results/reportField.do

Problem 1

- **Create and query a Dataset**
  - Navigate to project: 4-advanced-spark in Eclipse
  - Find and and open Problem01.java (src/main/java)
  - Look at // TODO for the problem description
  - Insert your code after the // TODO and run it

Problem 2

- **Create and join two datasets**
  - Navigate to project: 4-advanced-spark in Eclipse
  - Find and and open Problem02.java (src/main/java)
  - Look at // TODO for the problem description
  - Insert your code after the // TODO and run it

Problem 3

- **Create a new dataset and the query the dataset**
  - Navigate to project: 4-advanced-spark in Eclipse

  - Complete the code in UnitCellExtractorProblem03.java
  - Complete the code in Problem03.java
  - Then run Problem03.java