Science For A Better Life

Single Point of Entry

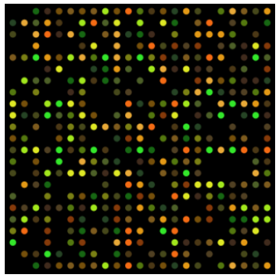
Integrating relational and semi-structured data with PostgreSQL

Dr. Ernst-Georg Schmid / NoSQL, Cologne / April 2014

The Pink Pineapple Problem

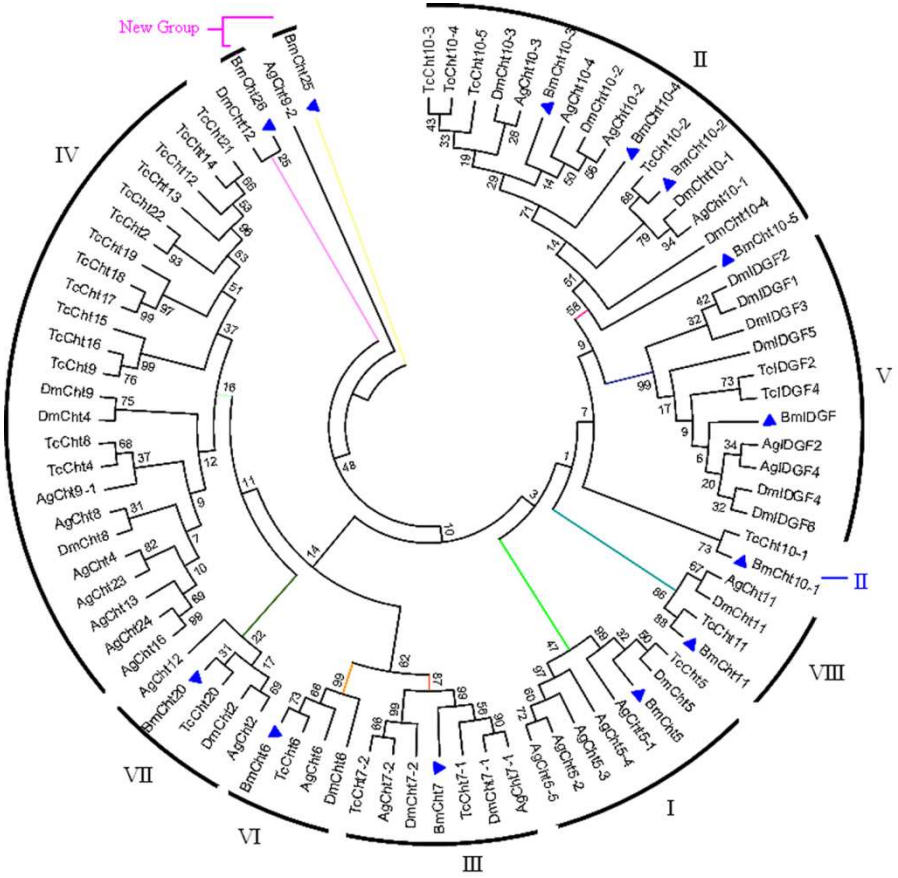
The Pink Pineapple Problem

…AGCTTTGACCTAGAGATCCGT
CAATGAGCTAGGCCTACATGAA
AGCTTTGACCTAGAGATCCGT
CAATGAGCTAGCTTATCT…

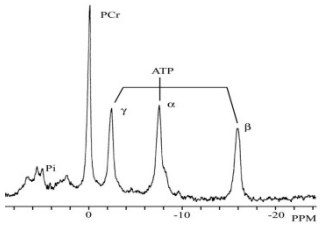


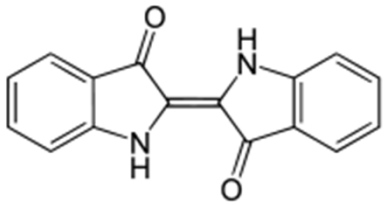
A	B	C
5.2345374	3.2543526	0.213541
sigma+1	alpha	rho
2.2	1	0.75

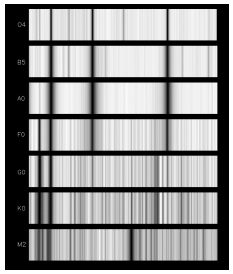
The Pink Pineapple Problem



The Pink Pineapple Problem



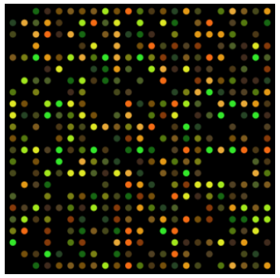




A	B	C
5.2	A	B	C
sig	5.2	A	B	C
2.2	sig	5.2345374	3.2543526	0.213541
	2.2	sigma+1	alpha	rho
		2.2	1	0.75

The Pink Pineapple Problem

…AGCTTTGACCTAGAGATCCGT
CAATGAGCTAGGCCTACATGAA
AGCTCCTGACCTAGAGGTCCGA
CAATGAGCTAGCTTATCT…



A	B	C
5.2		

A	B	C
5.2		
sig		
2.2		

A	B	C
5.2		
sig		
2.2		

A	B	C
5.2345374	3.2543526	0.213541
sigma+1	alpha	rho
2.2	1	0.75

The Pink Pineapple Problem

…AGCTTTGACCTAGAGATCCGT
CAATGAGCTAGGCCTACATGAA
AGCTCCTGACCTAGAGGTCCGA
CAATGAGCTAGCTTATCT…

…AGCTTTGACCTAGAGATCCGT
CAATGAGCTAGGCCTACATGAA
~~AGCTTTGACCTAGAGATCCGT~~
CAATGAGCTAGCTTATCT…

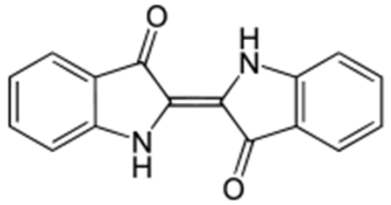
…AGCTTTGACCTAGAGATCCGT
CAATGAGCTAGGCCTACATGAA
AGCTCCTGACCTAGAGGTCCGA
CAATGAGCTAGCTTATCT…

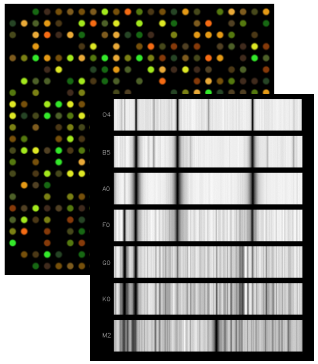
Diverse data along the way

…AGCTTTGACCTAGAGATCCGT
CAATGAGCTAGGCCTACATGAA
AGCTTTGACCTAGAGATCCGT
CAATGAGCTAGCTTATCT…

Text

Undirected cyclic graphs

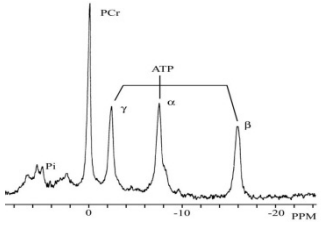




Uncompressed images

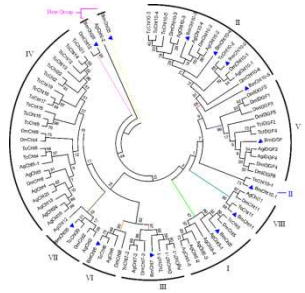
Sets of mixed types

A	B	C
5.2	A	B	C
sig	5.2	A	B	C
2.2	sig	5.2345374	3.2543526	0.213541
	2.2	sigma+1	alpha	rho
		2.2	1	0.75



Number arrays

Trees

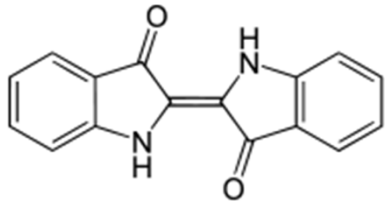


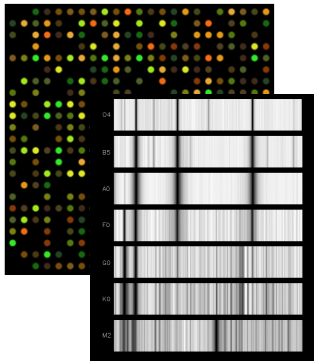
Growing data along the way

…AGCTTTGACCTAGAGATCCGT
CAATGAGCTAGGCCTACATGAA
AGCTTTGACCTAGAGATCCGT
CAATGAGCTAGCTTATCT…

600 GB
per sequencer
per week
+300TB
per year

10 million
+600000
per year





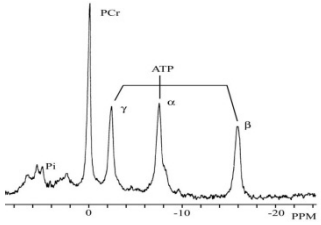
300TB
+300TB
per year

?

A	B	C
5.2		
sig		
2.2		

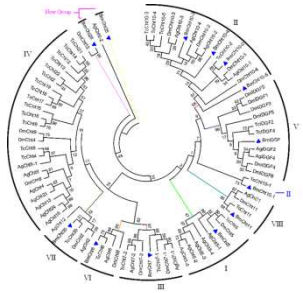
A	B	C
5.2		
sig		
2.2		

A	B	C
5.2345374	3.2543526	0.213541
sigma+1	alpha	rho
2.2	1	0.75



?

> 1 million
nodes



Heterogenous storage

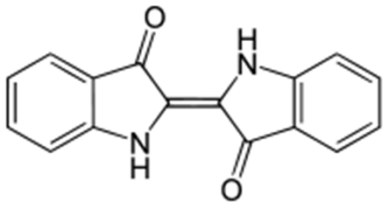
…AGCTTTGACCTAGAGATCCGT
CAATGAGCTAGGCCTACATGAA
AGCTTTGACCTAGAGATCCGT
CAATGAGCTAGCTTATCT…

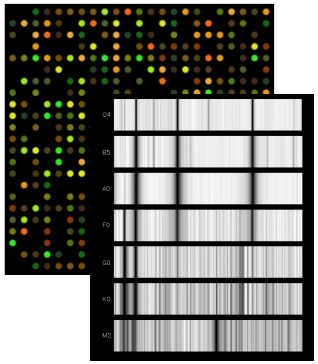
mongoDB

i·R·O·D·S

elasticsearch.

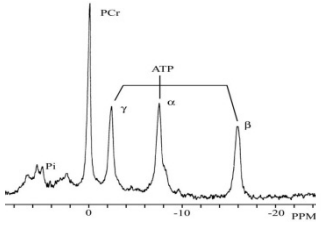
ORACLE





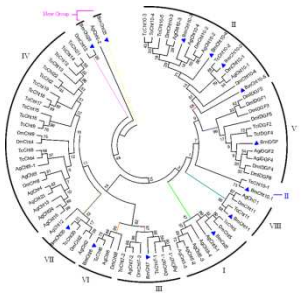
SciDB

A	B	C
5.2	A	B	C
sig	5.2	A	B	C
2.2	sig	5.2345374	3.2543526	0.213541
	2.2	sigma+1	alpha	rho
		2.2	1	0.75



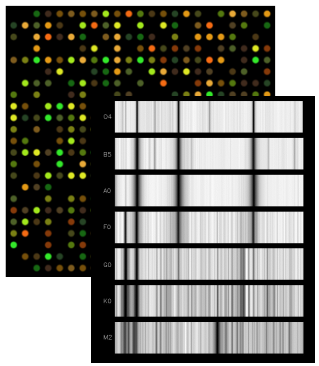
SciDB

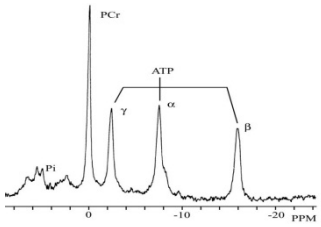
ORACLE

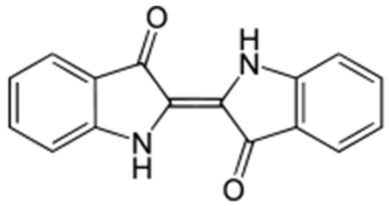


Problem centric integration

…AGCTTTGACCTAGAGATCCGT
CAATGAGCTAGGCCTACATGAA
AGCTTTGACCTAGAGATCCGT
CAATGAGCTAGCTTATCT…

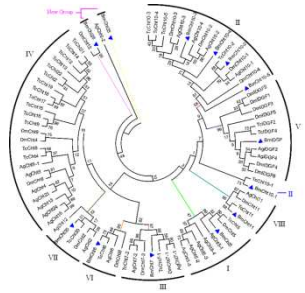






A	B	C
5.2		
sig		
2.2		

A	B	C
5.2345374	3.2543526	0.213541
sigma+1	alpha	rho
2.2	1	0.75



Problem centric integration

Every organization that tried to integrate all of this data in a single database…

…has failed!

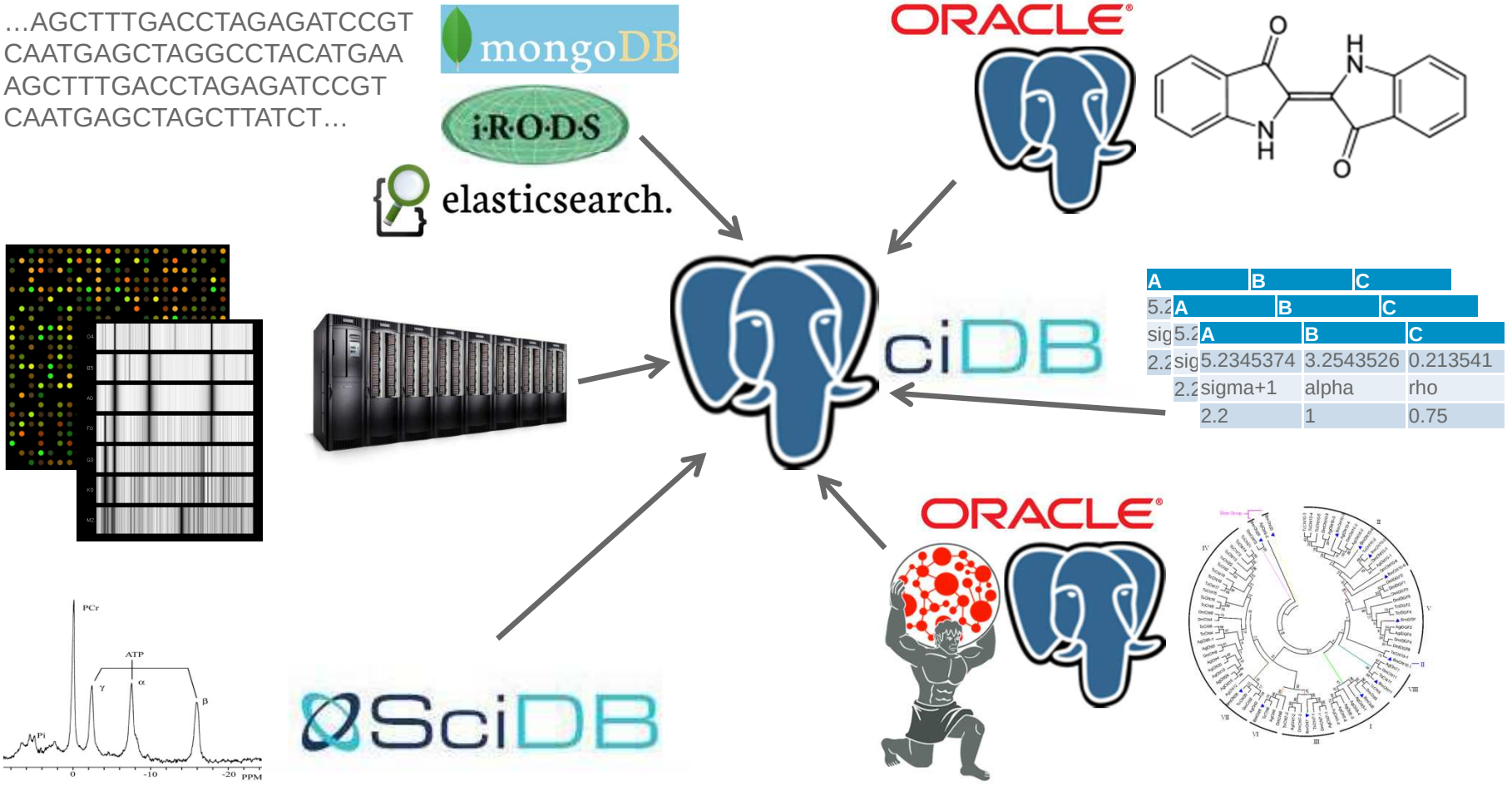
Problem centric integration

Still, some kind of integration is needed

- One system as Single Point of Entry
- SQL as lingua franca
- Broad driver support
- Permanently move data only when needed (e.g. for archival)
- „Projection first“ queries:

„Move computation, not data!“

PostgreSQL


…AGCTTTGACCTAGAGATCCGT
CAATGAGCTAGGCCTACATGAA
AGCTTTGACCTAGAGATCCGT
CAATGAGCTAGCTTATCT…
mongoDB
i·R·O·D·S
elasticsearch.
ORACLE
ciDB
SciDB
ORACLE
A B C
5.2 A B C
sig 5.2 A B C
2.2 sig 5.2345374 3.2543526 0.213541
2.2 sigma+1 alpha rho
2.2 1 0.75
PCr
ATP
Pi
PPM

Why PostgreSQL?

Because it‘s the world‘s most advanced open source database. ;-)

And here are some less bragging reasons…

XML

XML is a datatype to store XML data in a database column

- Checks for well-formedness
- Type-safe operations

http://www.postgresql.org/docs/9.3/static/datatype-xml.html

XML Support is output-biased

- Relation to XML mapping functions with matching XML Schema generation
- Checks for *xml_is_well_formed*, *IS DOCUMENT*, *XMLEXISTS*
- XPath 1.0 queries
- Aggregation function

- Validation only with DTD, no XML Schema support
- XML cannot be transparently indexed
- No comparison operators
- Memory intensive
- Encoding sensitive

JSON

JSON is a datatype to store JSON data in a database column

- Checks for valid JSON
- Type-safe operations

http://www.postgresql.org/docs/9.3/static/datatype-json.html

JSON

JSON support is almost complete

- Relation to JSON and JSON to relation mapping functions
- JSON object manipulation functions
- JSON object access operators
- Aggregation function

- JSON cannot be transparently indexed

HSTORE

HSTORE is a self-contained key/value store in a database column

- Key and value are strings
- Keys are hashed
- Arbitrary number of k/v pairs per HSTORE
- Keys are unique per HSTORE
- Values can be NULL

http://www.postgresql.org/docs/9.3/static/hstore.html

HSTORE

HSTORE support is complete

- Relation to HSTORE and HSTORE to relation mapping functions
- HSTORE key, value manipulation functions
- HSTORE access operators
- HSTORE to JSON mapping
- HSTORE can be transparently indexed

FDW + NoSQL

Foreign Data Wrappers map external data sources transparently into PostgreSQL

- Supports SELECT, INSERT, UPDATE, DELETE
- Datatype conversion
- Table size estimation
- Provide ANALYZE statistics
- Provide EXPLAIN information
- Impose user/role/privilege security

http://www.postgresql.org/docs/9.3/static/sql-createforeigndatawrapper.html
http://www.postgresql.org/docs/9.3/static/fdwhandler.html

CSTORE

CSTORE provides a columnar data store

- Implemented as FDW
- Column compression
- Column projection
- Full PostgreSQL integration on datatype and optimizer level

http://www.citusdata.com/blog/76-postgresql-columnar-store-for-analytics

JSONB

JSONB is the fusion of JSON and HSTORE

- Checks for valid JSON
- Type-safe operations
- Full transparent indexing
- Complete semi-structured document store in PostgreSQL
- Comes with PostgreSQL 9.4 (3rd Quarter 2014)

http://obartunov.livejournal.com/177247.html

Science For A Better Life

Single Point of Entry

Integrating relational and semi-structured data with PostgreSQL

Dr. Ernst-Georg Schmid / NoSQL, Cologne / April 2014