BIO4J

PABLO PAREJA

GRAPHDEVROOM 2014

WHAT IS BIO4J

IN ONE SENTENCE

Bio4j is a bioinformatics *graph*-based data platform **integrating** most data available in the most representative **open data sources** around **protein information** available today.

DATA

- *UniProt KB* (SwissProt + Trembl)
- *Gene Ontology* (GO)
- *UniRef* (50,90,100)
- *RefSeq*
- *NCBI taxonomy*
- *Expasy Enzyme DB*

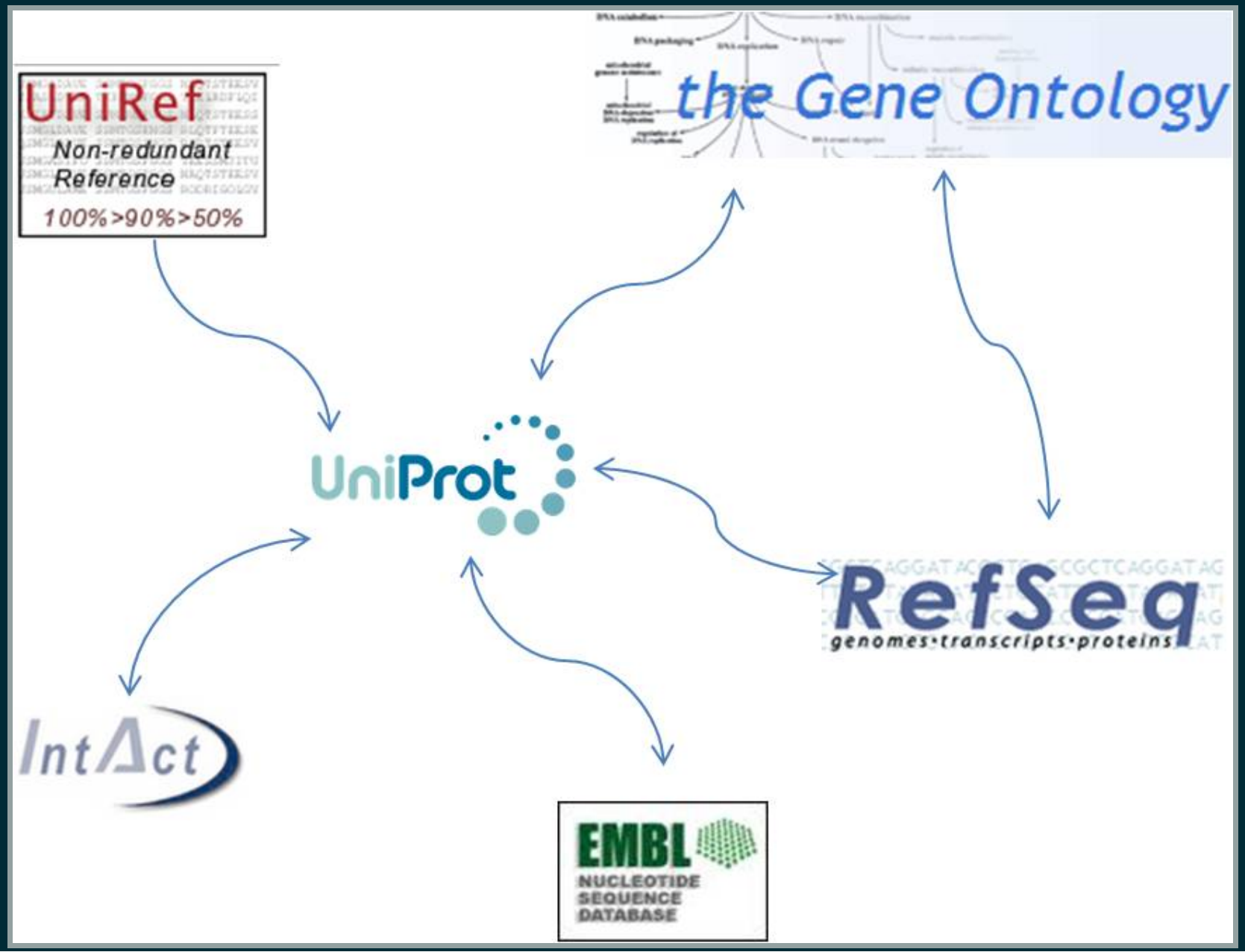
OPEN!

- **code** AGPLv3
- **data** integrates only open data
- **implementation & release** process is 100% public and totally **transparent**

WHY BIO4J?

BIOLOGY & DBS TODAY

Highly **interconnected** overlapping knowledge **spread** through *different databases*


UniRef
Non-redundant
Reference
100%>90%>50%
the Gene Ontology
UniProt
RefSeq
genomes·transcripts·proteins
IntAct
EMBL
NUCLEOTIDE
SEQUENCE
DATABASE

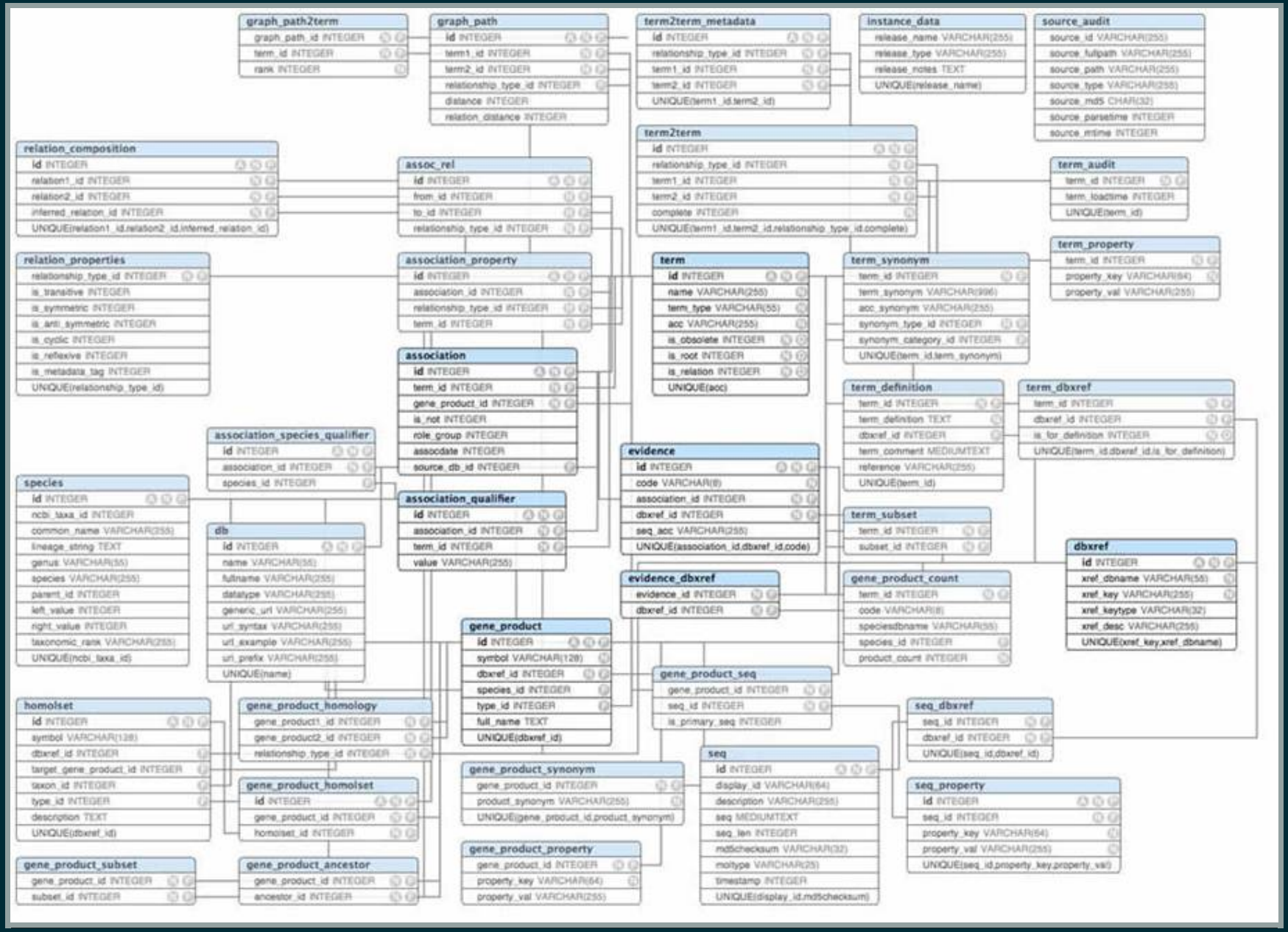
WHY GRAPHS

In most cases all data is modeled in or sometimes even just as plain *CSV* .

WHY GRAPHS

That might be OK for simple scenarios but as the **amount** and **diversity** of **data grows**, **domain models** become crazily **complicated**!

Doesn't look very compelling right? :)



WHY GRAPHS

With a relational paradigm the double implication

Entity <–> **Table**

does not go both ways

NOT-SO-GOOD IMPLICATIONS

- **Auxiliary tables**
- **Artificial IDs**
- Dealing with **raw tables** (in spite of Entity-relationship diagrams)
- **Integrating** new knowledge becomes **difficult**

BIOLOGY /= TABLE

Life in general and **biology** in particular are probably not 100% like a graph…

but one thing's sure, they

WHY GRAPH DATABASES

- **Data** stored in a way that **semantically represents its own structure**
- Incorporating new data is easy –> **scalability**

WHY GRAPH DATABASES

- **Vertex-centric** *(local)* indices allow for complex traversals –> overcoming **supernode problem**

CLOUD

- data as a service
- machine configurations

DETAILS ABOUT BIO4J

A BIT OF HISTORY

From the beginnings to the BigData platform it is today

HOW IT ALL STARTED

- Need for **massive access** to *Gene Ontology* annotations
- **BG7** bacterial genome annotation system
- Need for massive direct access to **protein information**

MORE AND MORE DATA!

- As *other* data sources were becoming a *bottleneck* they were being added to Bio4j
- First it was Uniprot KB, then Uniref and **we didn't stop yet :)**

NUMBERS

- 10^9 edges
- 2×10^8 nodes
- 6×10^8 properties
- 150 edge types
- 40 node types

BIO4J STRUCTURE

Bio4j importing process is **modular** and **customizable** allowing you to import just the data you are interested in.

DATA SOURCES - MODULES I

- Gene Ontology (GO)
- ExPASy Enzyme DB
- RefSeq

DATA SOURCES - MODULES II

- UniRef -> 50, 90, 100
- NCBI taxonomy tree -> GI index
- Uniprot KB -> Swissprot/Trembl, interactions…

DATA SOURCES - MODULES III

Just keep in mind that you must be **coherent**

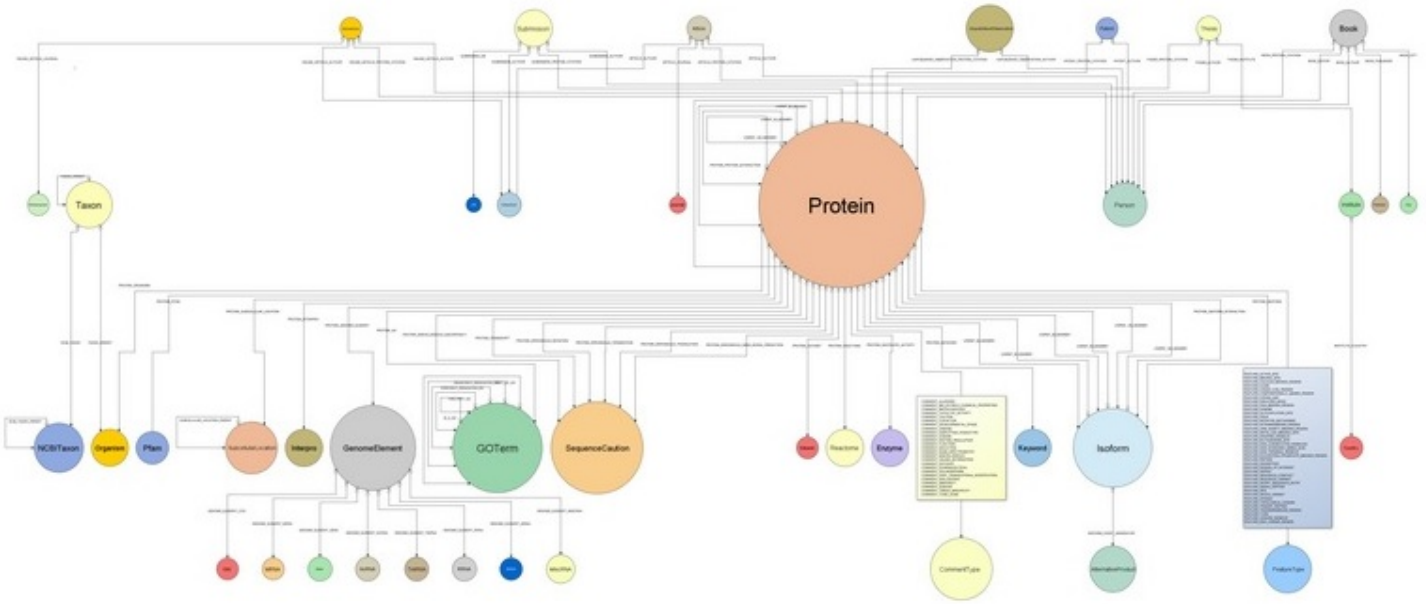
e.g. you cannot import protein interactions if you didn't import any protein yet!

BIO4J APIS

1. abstract **domain model**
2. **Blueprints** implementation
3. **technology-specific** versions

DOMAIN MODEL

Bio4j database has a **well-defined** domain model and all nodes and relationships comply with this abstract model


Protein
Taxon
NCBITaxon
Organism
Pfam
Interpro
GenomeElement
GOTerm
SequenceCaution
Enzyme
Keyword
Isoform
Person
Book

WHY DOMAIN MODEL?

- abstract over Blueprints
- more precise **typing**
- implementations can use technology-specific features

KEY ADVANTAGE

Different **graph topologies** at the storage level, *same* **domain model.**

Example: use **type nodes** in *Titan*, **labels** in *Neo4j*.

BLUEPRINTS LAYER

A default Blueprints implementation of the abstract model.

Apart from the set of interfaces developed as the **first layer** for the *domain model* there's an **extra layer** that uses *Blueprints*. This way we're going one step further for making the domain model **independent** from the choice of *database technology*

TECHNOLOGY-SPECIFIC

Optimizations, features, etc.

- Neo4j
- Titan (WIP)
- OrientDB (planned)

WHY NEO4J

- wide **adoption**
- stable
- **Cypher**

WHY TITAN

- **local!** indexes
- **on-disk** access
- **type** definitions -> *constraints!*

BIO4J AND THE CLOUD

- **Interoperability** and data distribution
- **Backup** and **storage**
- **Scalability**
- Applications and service providers on the cloud
- Cost-effective

DEV AND RELEASE PROCESS

- coordinate **data** and **code**
- Semantic Versioning
- **Cloud** integration, distribution, deployment, …

HOW?

- Statika cloud, data + code, modules (see next talk)
- sbt build Java + Scala, automated Bio4j-specific test & release
- git + github versioning, docs, collaboration, coordination

HOW TO USE BIO4J?

HOW WE USE IT

- bg7 genome annotation
- **mg7** metagenomics analysis
- comparative genomics, network analysis, genome assembly, …

CASE STUDY II

Ohio State University

- **Integration** and **analysis** of Chip-seq data
- **Modeling** genomic information and **gene regulatory networks**

CASE STUDY III

Berkeley Phylogenomics Group

- Graph database for *Big Data challenges* in **genomics** developed **on top of Bio4j**

COMMUNITY

- @bio4j twitter
- bio4j github org
- bio4j-user google group
- bio4j linkedin

WHO'S DOING BIO4J?

OH NO SEQUENCES!

Era7 bioinformatics R&D group

- **web ->** ohnosequences.com
- **Github ->** ohnosequences

TEAM

- Pablo Pareja
  project leader & main dev
- Eduardo Pareja-Tobes
  technology & architecture
- Raquel Tobes
  bio data integration

TEAM

- Alexey Alekhin
  Statika, release process, dev
- Marina Manrique
  bio data integration
- Evdokim Kovach
  dev