Bio4j

Pablo Pareja

February 17, 2014

ohnosequences!

era7 bioinformatics

0.0.1 *what* is Bio4j

0.0.2 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.

0.0.3 data

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

0.0.4 open!

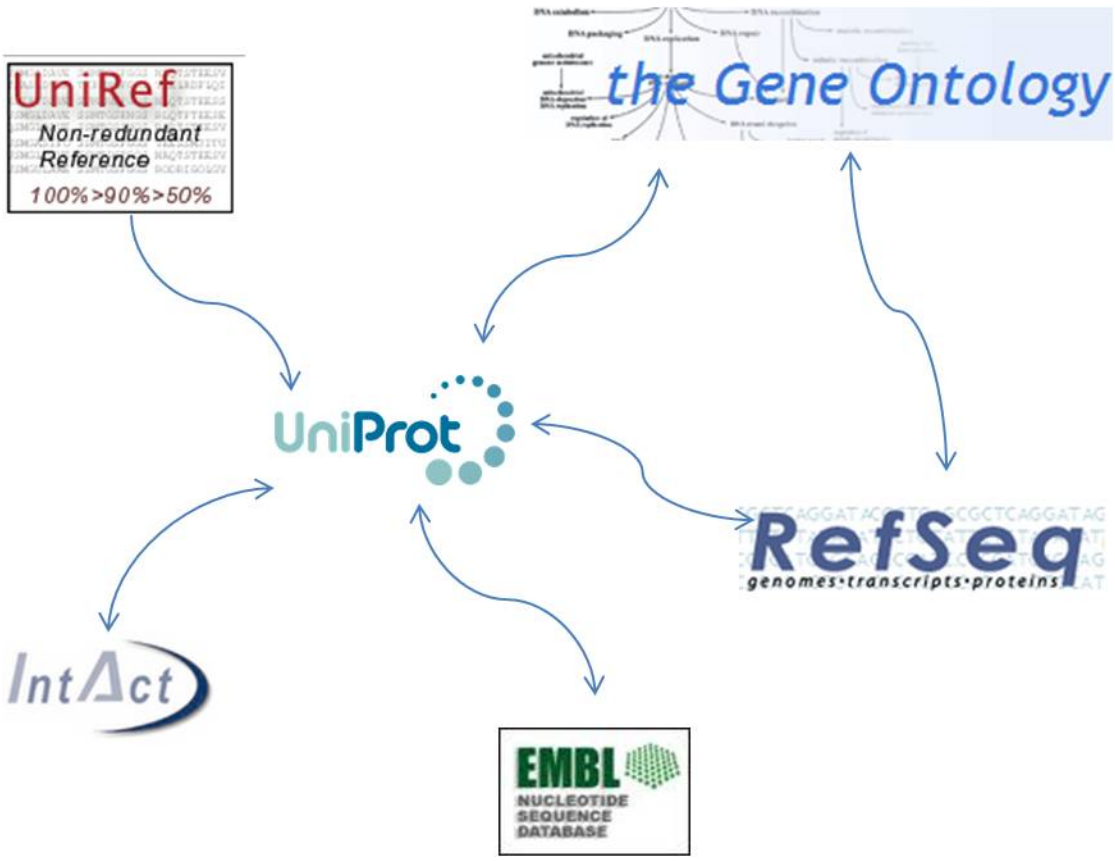
- **code** AGPLv3
- **data** integrates only **open data**
- **implementation & release** process is 100% public and totally **transparent**

0.0.5 *why* Bio4j?

bio data + graph databases + the cloud

0.0.6 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

0.0.7 why graphs

In most cases all data is modeled in ***Relational Databases*** or sometimes even just as plain **csv** ***files***.

0.0.8 why graphs

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

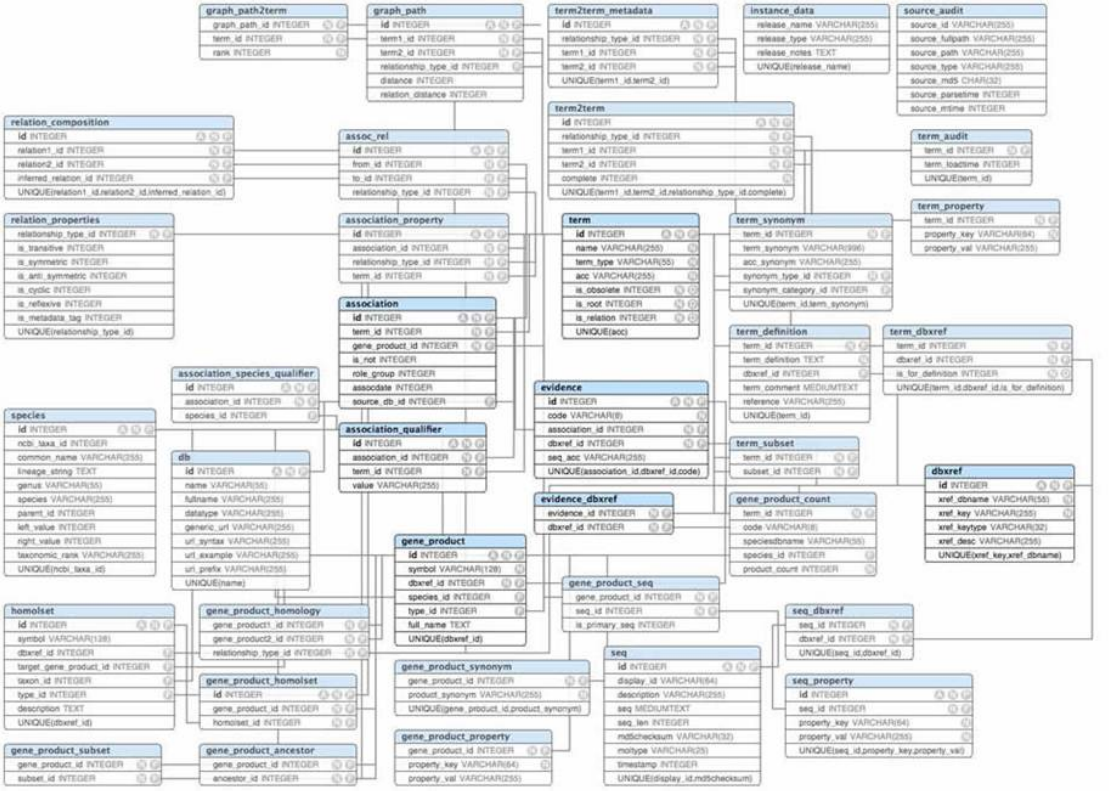


Figure 1: *Doesn't look very compelling right? :)*

0.0.9 why graphs

With a relational paradigm the double implication

Entity <–> **Table**

does not go both ways

0.0.10 *not-so-good* implications

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

0.0.11 biology *!=* table

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

but one thing's sure, they ***are not a set of tables!***

0.0.12 why graph databases

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

0.0.13 why graph databases

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

0.0.14 cloud

- data as a service
- machine configurations

0.0.15 *details* about Bio4j

data, model, technologies, APIs...

0.0.16 a bit of *history*

From the beginnings to the BigData platform it is today

0.0.17 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**

0.0.18 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 :)**

0.0.19 numbers

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

- 40 node types

0.0.20 Bio4j structure

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

0.0.21 data sources - modules I

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

0.0.22 data sources - modules II

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

0.0.23 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!

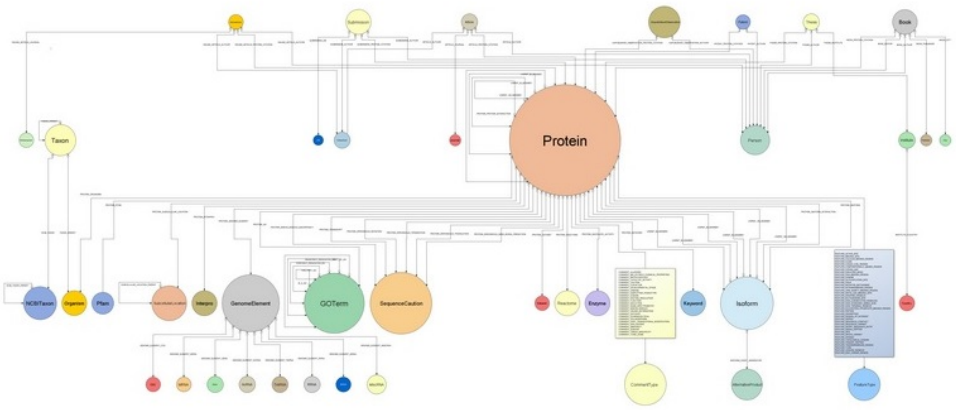
0.0.24 Bio4j APIs

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

0.0.25 domain model

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

---



0.0.26 domain model *why?*

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

0.0.27 Key advantage

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

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

0.0.28 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*

0.0.29 technology-specific

Optimizations, features, etc.

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

0.0.30 why Neo4j

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

0.0.31 why Titan

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

0.0.32 Bio4j and the cloud

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

0.0.33 dev and release process

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

0.0.34 how?

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

---

0.0.35 *how* to use Bio4j?

use cases, case studies, community

0.0.36 use cases

0.1

0.1.1 how we use it

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

0.1.2 case study II

Ohio State University

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

0.1.3 case study III

Berkeley Phylogenomics Group

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

0.1.4 community

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

0.1.5 *who*'s doing Bio4j?

research group, team

0.1.6 oh no sequences!

Era7 bioinformatics R&D group

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

0.1.7 team

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

0.1.8 team

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