A graph database to store and manage phenotypic, pedigree and genotypic data of livestock
Caveat

Premise:

One of the coauthors left (the main expert ... argh!)

“Salvage-whatever-possible”

Tell you what this is about

Tell you how it basically works

Tell you what we have so far set up

Won’t be able to show any comparison in performance (it was planned but ...)

Doesn’t mean this can’t be taken up again (maybe someone is interested!)
Text files & spreadsheets

<table>
<thead>
<tr>
<th>country</th>
<th>cows</th>
<th>Pedigree</th>
<th>records</th>
</tr>
</thead>
<tbody>
<tr>
<td>Germany</td>
<td>~ 4 million</td>
<td>~ 20 million</td>
<td>&gt; 300 million</td>
</tr>
</tbody>
</table>

- parse each time (e.g. by chromosome, subset of animals etc ...)
- less stable and less safe
- increasingly impractical with big data (e.g. HD SNP-chips, whole-genome sequences etc ...)
Relational databases

- Relational databases
  - Tables
  - Relations (between tables)
- SQL (queries instead of writing a script)

```
SELECT snp_genotype FROM genotypes_table WHERE chromosome=1;
```

1. more practical
2. faster
3. safer and more stable
Graph theory

- **Euler** and the 7 bridges of Königsberg

- “*Solutio problematis ad geometriam situs pertinentis*”. Commentarii Academiae Scientiarum Imperialis Petropolitanae 8 (1736) 128-140.

- **Graphs: nodes and connections**
  - common models for natural and human structures
    - computer networks
    - molecules (atoms and chemical bonds)
    - habitats and migration paths (breeding patterns, spread of diseases or parasites …)
Graph database

- **nodes** (e.g., movies, directors, actors), **connections** (e.g., cast, direction) and **properties** (e.g., character name, actor name, movie year ...)

**Features (promises):**
- no indexes
- suited for associative data
- faster (at least for graph-like queries)
- naturally translates into object-oriented programming
- scales better to large datasets
- accommodates better changing data
Our graph database

Working example

- animal dataset (buffaloes): phenotypes, genotypes and pedigree
- 3 layers: traits, animals, markers

Milk yield

[3322 kg]

rs123456

chr: 1
pos: 325326

traits

animals (kinship value on pedigree connections)

markers
Setting up the graph database

- Graph DBs are still an active field of research
- Not yet a standard query language (like SQL for relational databases)
- Lack of mature commercial products and user-friendly interfaces
- Several different ongoing projects

  - Ruby (create and populate the graph database)
  - Json (format for data interchange – associative array or hash)
Web Interface

Neo4j web administration

Server URL: http://localhost:7474
Kernel version: Neo4j - Graph Database Kernel 1.8.0.06

For more information, help and examples, please visit the Neo4j community site.

More about Charts
More about KPIs

47 nodes
179 properties
206 relationships
7 relationship types

Graph chart showing nodes, properties, and relationships over time.
Ruby gem to traverse the graph

```ruby
n1 = Neography::Node.load(15)
qu1 = n1.outgoing(:contain_maker).depth(2).include_start_node
```

```
#<Neography::NodeTraverser::0x686baa51 @order="depth first", @filter={"language"=>"builtin", "name"=>"all"}, @relationships=[{"type"=>"contain_maker", "direction"=>"out"}], @depth=2, @uniqueness="none", @from=#<Neography::Node position="85031448", name="AX-85040742", chromosome="15">>
```

Marker name: AX-85040742
Marker chromosome: 15
Marker position: 85031448

[parse the hash with Ruby]
Cypher

• declarative query language to search and update the graph (no need to traverse
  the graph structure writing a script)
• still growing and maturing.
• some keywords (e.g. WHERE, ORDER BY) are inspired by SQL

START animal=node:node_auto_index(ID = ‘ITM123456789’)
MATCH animal[:related]->()-[:related]->related_to
RETURN animal, related_to

START a=node(4) RETURN a
Graph visualization
Breeding and genomics data are associative

Scaling is an issue (big data)

Graph databases may offer a convenient alternative to store and manage data in animal breeding and genomics

Relatively recent area of research

R&D needed before a “commercial” product can be obtained

Compare with relational DBs in terms of performance, ease of use etc …