KATHOLIEKE UNIVERSITEIT LEUVEN

DistriNet RESEARCH GROUP

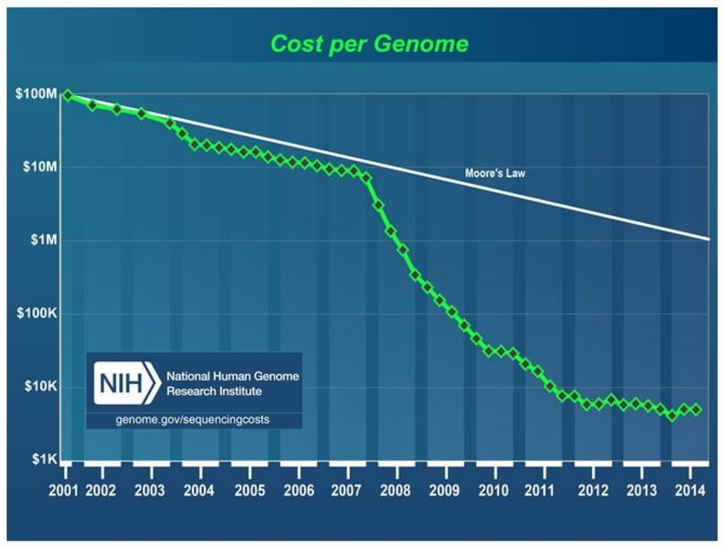
B. Gosselé
Prof. Dr. R. Wuyts

Exascience Life Lab

# Schaalbare genoomanalyse

https://github.com/bgossele/geminicassandra

## Context




Wetterstrand KA. DNA Sequencing Costs: Data from the NHGRI Genome Sequencing Program (GSP) Available at: www.genome.gov/sequencingcosts. Accessed 15/12/2014

Vraag naar efficiënte, schaalbare analyse en opslag van gegevens ↗↗

## Oplossing

- Dataschema
  - Genotype-kolommen => kolom voor elke sample

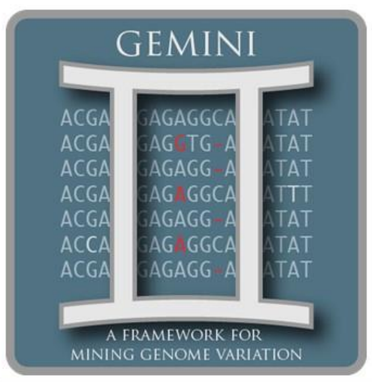
| variant_id | ref | alt | … | gt_type_alex | gt_type_john | … | gt_depth_alex | gt_depth_john | … |
|---|---|---|---|---|---|---|---|---|---|

  - Arbitraire queries:
    - Zonder indices
    - Duplicatie data: extra tabellen met geschikte keys definiëren

| sample_name | gt_depth | variant_id | chrom | start | variant_id | sex | sample_name |
|---|---|---|---|---|---|---|---|

  - Extra tabellen verbergen voor gebruiker
  - Geen joins => denormalizatie
- Inladen genoomdata
  - Vullen variants-tabel efficiënt te parallelliseren
  - Cores schrijven parallel nr 1 Cassandra cluster
    >< SQLite: elk nr 1 DB, achteraf mergen

## Context: case study




- Multi-sample genoomanalyse
- Annotaties, vb:
  - aaf_1kg_afr (frequentie genetische variant in Afrikaanse populatie)
  - is_somatic (somatische mutatie, mogelijk carcinogeen)
- Ad-hoc queries
- SQLite = te traag, schaalt niet

```
$ gemini query --header \
               -q "SELECT chrom, start, end, ref, alt, gene, (gts).(*) \
                   FROM variants" \
               --gt-filter "(gt_types).(phenotype==1).(==HOM_REF).(all) \
                            and \
                            (gt_depths).(phenotype==2).(==HOM_REF).(none)" \
               extended_ped.db
```

## Oplossing

- Queries
  - Query op hoofdtabel (`variants, samples`) => splitsen in subqueries op hulptabellen. Bvb:

```
SELECT * FROM variants WHERE chrom = 'chromX'
                         AND start > 1500 AND gt_type.alex = HET
=> SELECT variant_id FROM variants_by_chrom_start
                     WHERE chrom = 'chromX' AND start > 1500
+    SELECT variant_id FROM variants_by_samples_gt_type
                     WHERE sample_name = 'Alex' AND gt_type = HET
```

  - Resultaten subqueries = verzamelingen
  - Combineren met set-operaties
  - Resulterende rijen, kolommen opvragen uit hoofdtabel

| Query | Resultaat |
|---|---|
| $p$ AND $r$ | $res(p) \cap res(r)$ |
| $p$ OR $r$ | $res(p) \cup res(r)$ |
| NOT $p$ | $I \setminus res(p)$ |

## Probleem

- Relationeel dataschema GEMINI
  - variants-tabel
    - Genotype-kolommen : compressed binary vectors
  - samples-table: geslacht, herkomst, familierelaties

variants

| variant_id | chrom | start | … | gts | … |
|---|---|---|---|---|---|
| 1 | chromX | 24 | … | *A/A*<br>*A/G*<br>*T/T*<br>*T/C …* | … |
| 2 | chromY | 3541 | … | *T/T*<br>*T/T*<br>*T/C*<br>*A/G …* | … |

- Ad-hoc queries:
  - Standaard SQL-queries (onmogelijk op binary genotype-arrays)
  - Daarom: --gt-filter: gt_type.alex == HET
  - Wildcards : (COL).(SAMPLE_WILDCARD).(WILDCARD_RULE).(RULE_ENFORCEMENT)
    Bvb. (gt_types).(sex == 1).(== HET).(all)

## Oplossing

- --gt-filter wildcards
  - Lange ketens subqueries op genotype-hulptabellen
  - Evaluatie subqueries voor verschillende samples => parallelliseren
- User interface
  - Gebruiker moet hulptabellen kennen
  - SQL-syntax licht gewijzigd => parsen
    - Queries binnen 1 hulptabel: CQL syntax
    - Gekoppeld met && i.p.v. AND, ||, NOT

```
$ gemini query  -q "SELECT chrom, start, subtype FROM variants
                    WHERE chrom = 'chromX' AND start > 5600 AND gene = 'gene A'"
                --gt-filter "(gt_types).(phenotype==1).(==HOM_REF).(all)"
```

```
$ geminicassandra query -q "SELECT chrom, start, subtype FROM variants
                            WHERE chrom = 'chromX' AND start > 5600 && gene = 'gene A'"
                        --gt-filter "[gt_types].[phenotype==1].[==HOM_REF].[all]"
```

## Apache Cassandra i.p.v. SQLite

- Schaalbaarheid ✓
- Relationele tabellen => columnair model ✓
- Concurrency ✓
- CQL ~= SQL; Python API ✓

Maar…

- Minder support voor indexing, querying
  - Geen joins
  - Beperkingen op WHERE-clause
    - » Cassandra = gigantische multimap <keys, columns> -> values
    - » Niet itereren en alle rijen filteren, maar hash(key) berekenen -> map nr correcte rijen
    - » Enkel opeenvolgend opgeslagen rijen opvragen
    - » PRIMARY KEY ((col1,…), (col_x)*)
      partition key | clustering key
      bepaalt node in cluster | bepaalt volgorde rijen met == partition key
    - » Geen OR, NOT, !=,…

## Conclusies

- Doel: schaalbare genoomanalyse
- GEMINI = genoom-DB + uitgebreide ad-hoc SQL-queries
- Apache Cassandra: compromis
  - Schaalbaarheid vs. query-features
  - Dataduplicatie, denormalisatie
  - Queries in applicatie-laag
  - Parallellisatie
  - Flexibel schema

ExaScience life lab