Challenges of working with a large database of routinely collected health data: Combining SQL and R

Joanne Demmler¹, Caroline Brooks¹, Sarah Rodgers¹, Frank Dunstan², Ronan Lyons¹

1. Swansea University
   Health Information Research Unit
   College of Medicine
   Grove Building
   Singleton Park
   Swansea SA2 8PP

2. Cardiff University
   Department of Primary Care
   & Public Health
   Neuadd Meirionnydd
   Heath Park
   Cardiff CF14 4YS
HIRU and the SAIL database

- HIRU – the Health Information Research Unit
- SAIL – Secure Anonymous Information Linkage
- Main aim of HIRU is to realise the potential of electronically-held, routinely-collected, person-based data to conduct and support health-related studies
- The SAIL databank already holds over 1.9 billion anonymised and encrypted individual-level records, from a range of sources relevant to health and well-being
How can these data be made available for research?

- In accordance with the principles of Information Governance
- To ensure data security, integrity and quality
- To maintain data usefulness

**SAIL references:**


HIRU methodology

Demographic data only
Anonymisation process
Validated, anonymised data
Recombine
Crypt and load

Operational system
NHS Wales Informatics Service

Data Provider

Other recombined data

hiru (Blue C)

Construct
ALF
Validate

HIRU (Blue C)
All analysis is done within the SAIL gateway

- data analysts retrieve data through SQL code from DB2 databank on Blue C replacement servers
- researchers analyse data using SPSS, STATA or R

Files are moved into the gateway using a FTP client

- no internet access within the gateway

Files are requested out of the gateway through a review process

- screening for potentially identifiable data
Why use R?

- Running SQL queries and creating tables
  - users do have restricted command line access to DB2
  - no access to advanced SQL options such as procedures
  - Brilliant way to create multiple SQL tables, e.g. `for loop & paste` command

- Evaluation and pre-cleaning of raw data
  - no need to create temporary tables in SQL or copy query results into different software package

- Programming heavy analysis
  - biomarkers
  - data mining (RWeka)
Challenges when working with R and SAIL – PART 1

- **R packages**
  - have to be installed manually in the SAIL gateway
  - Possibility to open a single connection to a CRAN mirror

- **Computing power**
  - SQL uses computing power of Blue C replacement servers
  - R only has remote desktop properties (equals to 1 core of a Xeon 5550@2.67 GHz processor, with allocated memory of 2GB RAM per user)
  - There are plans to install R on a separate, very powerful server (a server each per statistics package: SPSS, STATA, R)
Connecting to SAIL with RODBC

1) Installation of ODBC driver
2) Installation of package RODBC in R
3) Start RODBC

```
library(RODBC)
```

4) Connect to SAIL (makes table views available)

```
channel <- odbcConnect("PR_SAIL")
```

5) Set up the WORKTMPT environment

```
odbcQuery(channel,"SET CURRENT SCHEMA = WORKTMPT")
```
Querying SAIL from R

Run a Query

```r
hw.table <- sqlQuery(channel, 
  
  SELECT DISTINCT a.ALF_E
  , a.GNDR_CD
  , b.EXAM_DT
  , TIMESTAMPDIFF(256, CHAR(TIMESTAMP_ISO(b.EXAM_DT) - TIMESTAMP_ISO(a.WOB)))
    as AGE_YRS
  , TIMESTAMPDIFF(64, CHAR(TIMESTAMP_ISO(b.EXAM_DT) - TIMESTAMP_ISO(a.WOB)))
    as AGE_MNTH
  , b.HEIGHT_CM
  , b.WEIGHT_KG
FROM WORKTMPT.JD_WECC_SUBSET_2 a
JOIN SAILCHDHV.EXAM b
ON a.CHILD_ID_E = b.CHILD_ID_E
WHERE TIMESTAMPDIFF(64, CHAR(TIMESTAMP_ISO(b.EXAM_DT) -
  TIMESTAMP_ISO(a.WOB))) >= 0
AND (HEIGHT_CM IS NOT NULL AND WEIGHT_KG IS NOT NULL)
AND GNDR_CD IN ('1', '2')
ORDER BY a.ALF_E, b.EXAM_DT;
```

Data retrieval: **R: 1:26 min, DB2 command line: 0:41 min, WinSQL: 3:32 min**

1,842,820 rows, 7 columns \(\rightarrow\) 602,975 individual children
Querying SAIL from SQL script in R

All SQL scripts have to be reviewed before data can be requested out of the gateway. It therefore makes sense to keep SQL scripts as separate files.

Run a query from an SQL script

```r
con <- file("hwcode.sql")
sql <- readLines(con)
sqlQuery(channel, paste(sql, collapse=" "))
close(con)
unlink("hwcode.sql")
```
## Create table in SAIL using SQL

**Create table**

```sql
> sqlQuery(channel, "CREATE TABLE WORKTMPT.JD_HW
(ALF_E BIGINT,
  GNDR_CD CHAR(1),
  EXAM_DT DATE,
  AGE_YRS SMALLINT,
  AGE_MNTH SMALLINT,
  HEIGHT_CM DECIMAL(31,8),
  WEIGHT_KG DECIMAL(31,8)
DISTRIBUTE BY HASH(ALF_E);
")
```

**Populate table**

```sql
> sqlQuery(channel, "INSERT INTO WORKTMPT.JD_HW
(SELECT DISTINCT a.ALF_E,
  a.GNDR_CD,
  b.EXAM_DT,
  TIMESTAMPDIFF(256,
  CHAR(TIMESTAMP_ISO(b.EXAM_DT) –
  TIMESTAMP_ISO(a.WOB))) as AGE_YRS,
  TIMESTAMPDIFF(64,
  CHAR(TIMESTAMP_ISO(b.EXAM_DT) –
  TIMESTAMP_ISO(a.WOB))) as AGE_MNTH,
  b.HEIGHT_CM,
  b.WEIGHT_KG
FROM WORKTMPT.JD_WECC_SUBSET_2 a
JOIN SAILCHDHV.EXAM b
ON a.CHILD_ID_E = b.CHILD_ID_E
WHERE TIMESTAMPDIFF(64,
  CHAR(TIMESTAMP_ISO(b.EXAM_DT) –
  TIMESTAMP_ISO(a.WOB))) >= 0
AND (HEIGHT_CM IS NOT NULL AND WEIGHT_KG
IS NOT NULL)
AND GNDR_CD IN ('1', '2')
ORDER BY a.ALF_E, b.EXAM_DT;
")
```

Create and populate table:
- **R:** 30 sec, **WinSQL:** 11 sec
Append data to SAIL table

```sql
sqlQuery(channel, "CREATE_INDEX WORKTMPT.JD_HW1_01 ON WORKTMPT.JD_HW (ALF_E)"
sqlQuery(channel, "ALTER TABLE WORKTMPT.JD_HW ADD COLUMN TEST CHAR(1)"
```

DB2 commands, which restructure the table (such as `reorg table`, `runstats`) will have to be run separately.

```system
system("db2 connect to PR_SAIL user xxx using xxx")
system("db2 reorg table WORKTMPT.JD_HW")
system("db2 runstats on table WORKTMPT.JD_HW with distribution and detailed indexes all")
system("db2 quit")
```
Investigating raw data

```r
hw.table <- sqlFetch(channel, "JD_HW")
```

OR

```r
hw.table <- sqlQuery(channel, "SELECT * FROM WORKTMPT.JD_HW")
```

Possible problems:
- typing errors
- wrong units (inches, feet / pounds, stones)
- serious congenital diseases (e.g. Dwarfism)
Removing impossible values

**Combined heights**

<table>
<thead>
<tr>
<th></th>
<th>before</th>
<th>after</th>
</tr>
</thead>
<tbody>
<tr>
<td>boys</td>
<td>1,842,820</td>
<td>1,795,606</td>
</tr>
<tr>
<td>girls</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Both height & weights **1,764,728** (96% of data)

**Combined weights**

<table>
<thead>
<tr>
<th></th>
<th>before</th>
<th>after</th>
</tr>
</thead>
<tbody>
<tr>
<td>boys</td>
<td>1,842,820</td>
<td>1,792,063</td>
</tr>
<tr>
<td>girls</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Filtering data against height and weight limits in R can be very time consuming **BUT** will be very fast in SQL on the supercomputer.
Saving data back to SAIL (*sqlSave*)

- can be slow
  - saving 1.7 million rows of data takes 2.5 hours (*fast=T* is 14 minutes quicker)

- might need special attention for very large tables
  - running out of internal memory or connection is timing out

- might need special attention to formatting of columns (e.g. 
  , *varTypes=c(EXAM_DT= "Date"),* decimals will be saved as double)

Best option to adhere with SAIL formatting conventions

- `create table` with *sqlQuery* and then use *sqlSave(..., rownames=F, fast=T, append=T)*
Conclusions

- R can successfully be used as an effective data processing & querying tool with SAIL

- R has added benefits, such as
  - evaluating data in the same application
  - automating queries
  - running DB2 commands over the command line

- When importing data from SAIL into the gateway the performance is dramatically reduced (need for separate, more powerful server)