Orchestrating privacy-protected non-disclosive omic data analyses in multi-cohort studies using DataSHIELD

Juan R Gonzalez
Bionformatics Research Group in Epidemiology
e-mail: juanr.gonzalez@isglobal.org

ISGlobal Barcelona Institute for Global Health

Institució CERCA Centres de Recerca de Catalunya

A partnership of:

"la Caixa" Foundation

CLÍNIC BARCELONA Hospital Universitari · Parc de Salut Barcelona · MAR · Universitat de Barcelona · upf. Universitat Pompeu Fabra Barcelona · Generalitat de Catalunya · Gobierno de España · Ajuntament de Barcelona · Fundación Ramón Areces

Microdata

Microdata = individual level data = individual patient data (IPD) [opposite to summarized data]

- Microdata are ***absolutely fundamental*** to contemporary science including biomedical, social and public health data

- For someone wishing to analyze, interpret and draw conclusions from data they provide
  - The only way to do certain analyses
  - Enhanced efficiency in some circumstances
  - Greater flexibility

- Microdata are *sensitive* and there are barriers to sharing

Constrains and barriers to sharing and combining microdata

- Ethico-legal and other governance restrictions (GDPR)
- Maintain control of intellectualy property
- Physical size of data
- **What should we do?**

"I'm sure there are better ways to disguise sensitive information, but we don't have a big budget."

The DataSHIELD approach

- ❑ Take "analysis to data" … not "data to analysis"
- ❑ Leave the data to be analyze on local servers behing the firewalls where they usually reside
- ❑ The analysis centre co-ordinates parallelized analyses in all studies simultaneously
- ❑ Tie analyses together with non-disclosive information
- ❑ Analytic processing – *and options for disclosure control* – located with the data

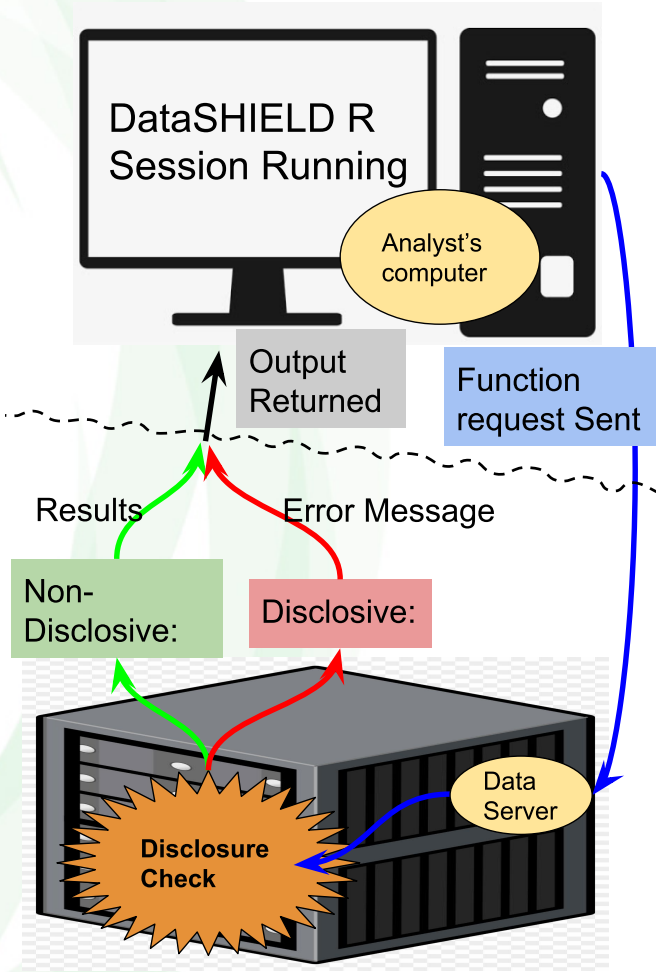
What is DataSHIELD?

- A privacy-preserving, non-disclosive, federated analysis software

- What it is
  - way of analyzing sensitive data, at a individual level, **without providing direct access** to original data

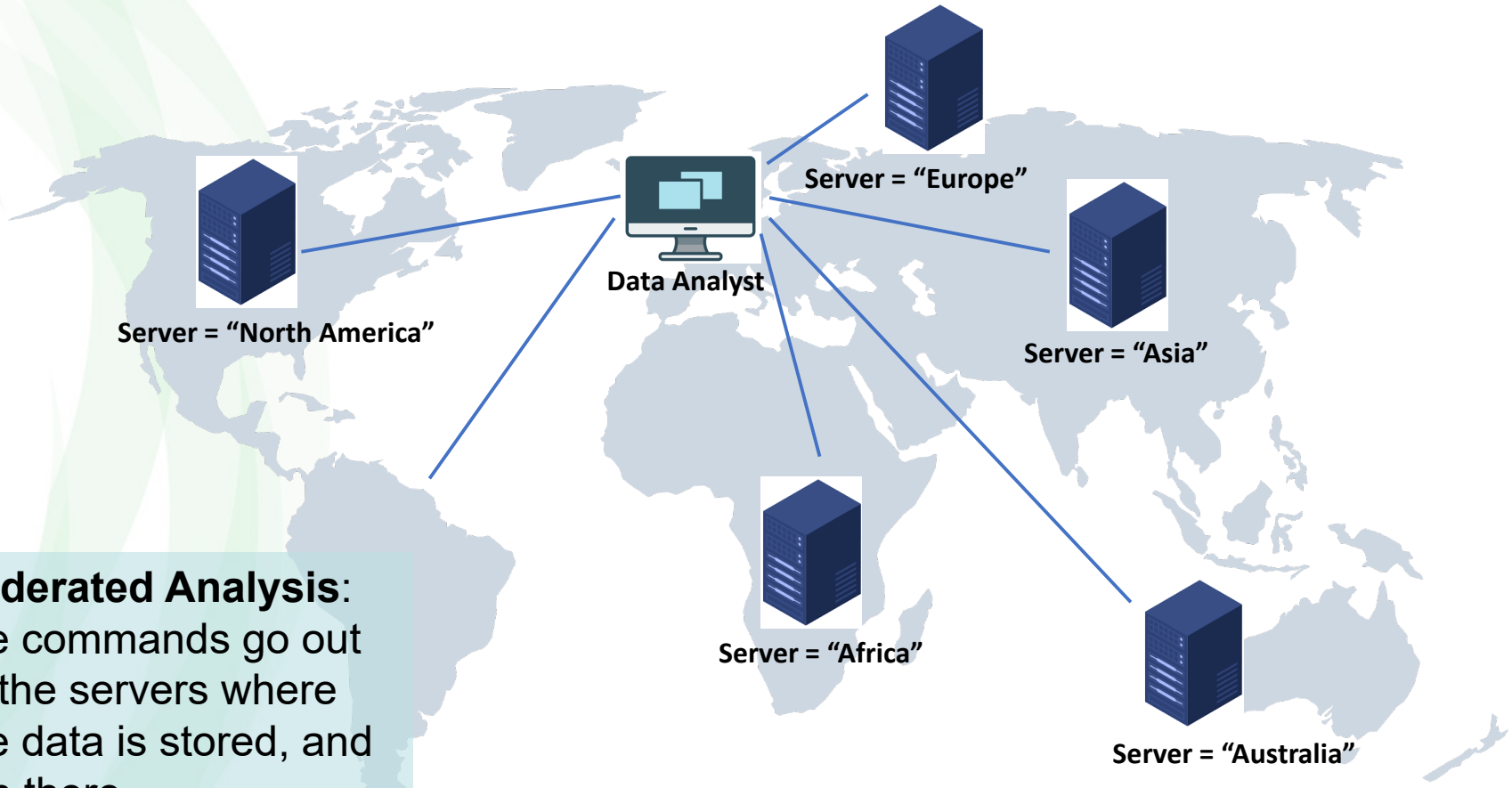
- What it is not
  - data harmonization platform

Privacy-preserving



- ❑ Functions are sent to data server
- ❑ Server runs function
- ❑ Built into function is disclosure check
- ❑ If disclosive, check will discover and only return error message, not results.

Federated analysis

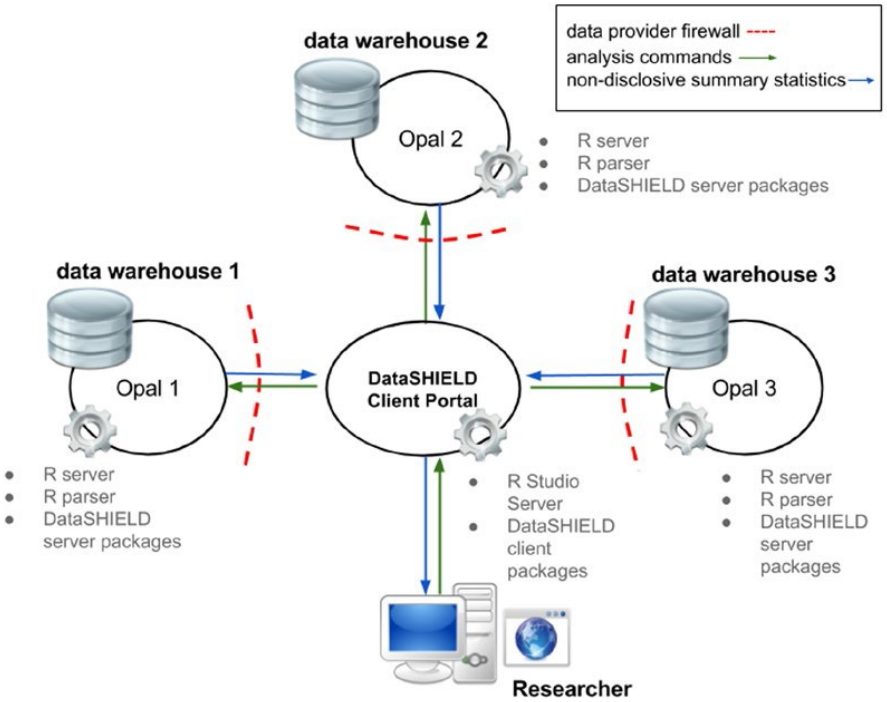


Federated Analysis: the commands go out to the servers where the data is stored, and run there.

DataSHIELD standard platform

Preliminary: data available **through Opal** (at owner institution)

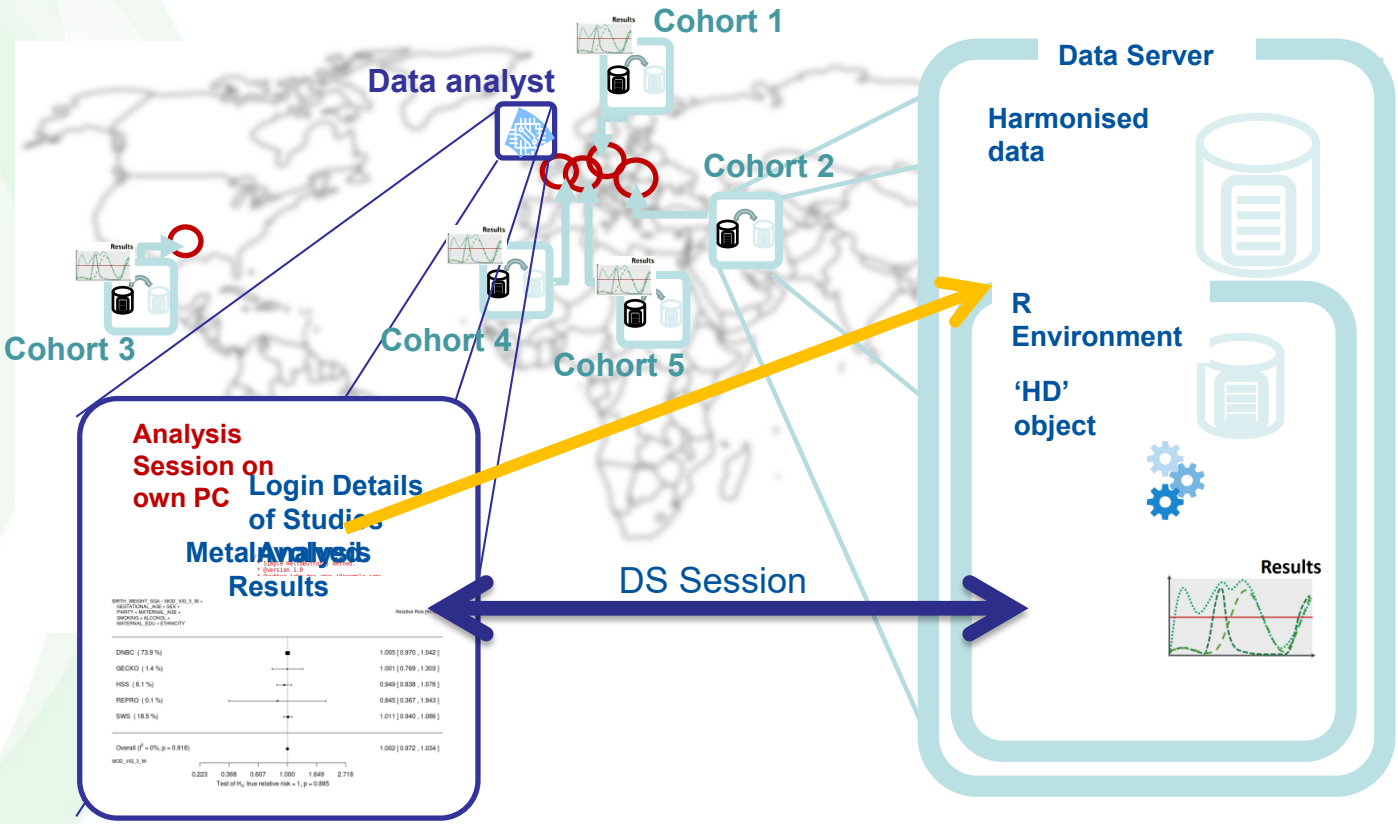
- ❑ Authenticate and authorize user
- ❑ Assign Opal table into the R server
- ❑ Execute DataSHIELD-verified R commands in R server from client side (data analyst)



Opal demo

https://opal-demo.obiba.org/ui/index.html

Animation of a DataSHIELD Analysis

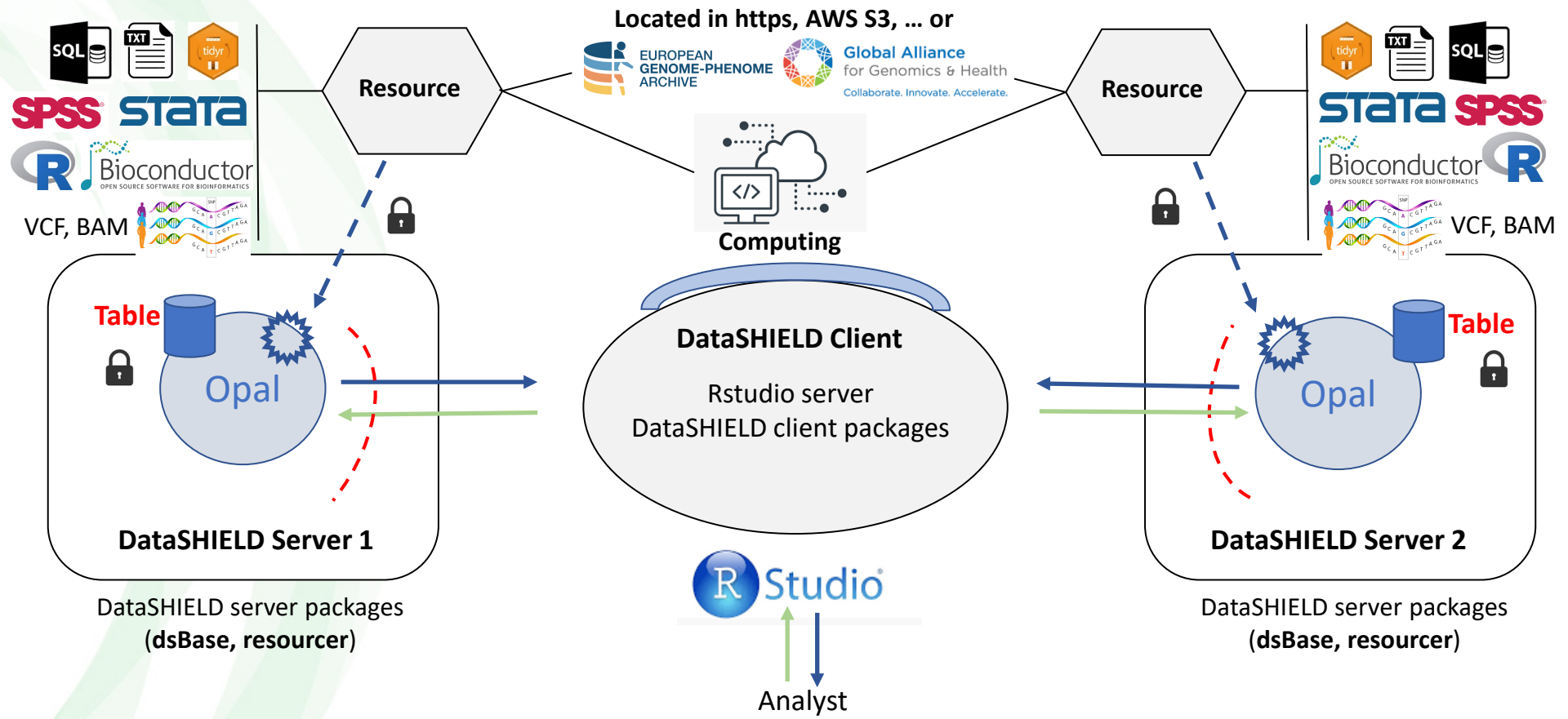


What king of analysis can we do?

- There are more than 100 functions within the DataSHIELD dsBaseClient package to choose from, in the latest released version (6.1). You can check the list with the base DataSHIELD functions here: https://data2knowledge.atlassian.net/wiki/spaces/DSDEV/pages/1184825438/List+of+all+DataSHIELD+functions+v6.1

- Also there is a wide range of other DataSHIELD Community packages for special type of analysis (e.g. causal effect analysis, analysis of omics data, analysis of geospatial data, etc.) which can be found here: https://www.datashield.org/help/community-packages

The resources: idea



Data store | R server | DataSHIELD Interface (DSI) | Data controller firewall | Non-disclosive statistics | DS analyst commands | Direct connection

Data Warehouse (Opal)

- Store data on an unlimited number of variables,
- Support MongoDB , Mysql , MariaDB and PostgreSQL as database software backend,
- Customized variable dictionaries,
- Import data from CSV, SPSS, SAS, Stata files and from SQL databases,
- Export data to CSV, SPSS, SAS, Stata files and to SQL databases,
- Incremental data importation,
- Connect directly to multiple data source software such as SQL databases and LimeSurvey ,
- Store data about any type of "entity", such as subject, sample, geographic area, etc.,
- Store data of any type (e.g., texts, numbers, geo-localisation, images, videos, etc.),
- Import and store genotype data as VCF files (Variant Call format),
- Advanced indexing functionality using ElasticSearch .

The resources: rational

- ❑ Use data at their original location (do not move/copy data when possible)
- ❑ Use data in their original format (no pre-processing, no loss of information)
- ❑ Use external computation facilities (no R limitations)

❑=> use DataSHIELD with large/big datasets (omics etc.) using their original format (Bioconductor, images, …)

The resources …

- are an alternative to Opal storage
- can be data resources stored in
  - files
  - or databases
  - or a remote application …
- can be computation resources
  - command executed locally or remotely
  - web services …

Resources in R

R package resource

- ❑ resource class
- ❑ ResourceClient class, connects to the data or computation resource
- ❑ ResourceResolver class, makes a ResourceClient from a resource

Source code: https://github.com/obiba/resourcer

Resources in R

ResourceClient is extensible

- FileResourceClient
  - File getter: local, https, file store (S3, GridFS, Opal…) etc.
  - File format interpreter: tidyverse (csv, spss, etc.), R data, VCF etc.
- SQLResourceClient
  - use DBI to connect to database: mysql, postgre, presto, spark etc.
- CommandResourceClient
  - ssh or shell
- …

The Resources

- coerce to **data.frame**, to fallback to standard DataSHIELD
- **dplyr** support, to analyse data in place
- access to **raw data**, for domain specific analysis (BioConductor)
- **delegate data analysis** to external command or web service

Resources examples

- CSV file (compressed or not)
- SQL database table
- R object stored in a R data file
- HL7 FHIR dataset
- GA4GH server
- EGA server
- HPC server accessible through SSH
- Python script
- Big data analytics system (Apache Spark, Dremio, …)
- Apps in docker images
- …

The resources …

Property	Description	Examples

url	Location of the resource	`https://example.org/some/file.rda`
		`file://path/to/file.csv`
		`ssh://example.org/work/dir?exec=plink`
		`mysql://dbhost:3306/mydb/mytable`

format	Data format (optional)	`SPSS` `ExpressionSet`
credentials	Data access (optional)	`token=Q3sDdsWq2dsx7` `username=user1 password=xxxxxx`

Not visible by DataSHIELD users

The Resources

Web standard : URL, **U**niform **R**esource **L**ocation

<scheme>://<authority><path>?<params>

Examples

https://github.com/isglobal-brge/brgedata/raw/master/data/gse66351_1.rda

mysql://192.168.2.12:3306/sim/CNSIM1

file:///srv/data/CNSIM2.zsav

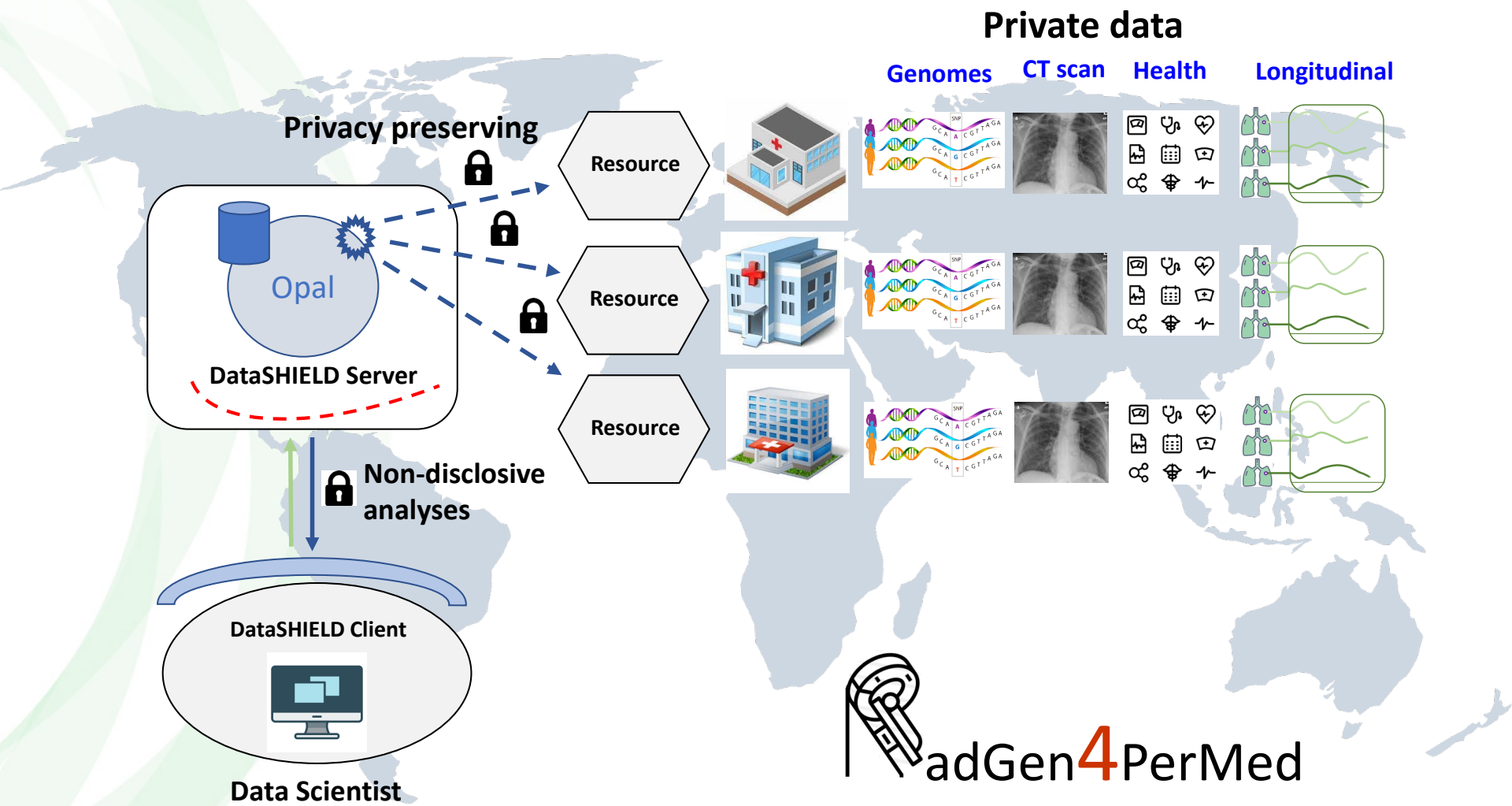
ssh://plink-demo.obiba.org:2222/home/master/brge?exec=ls,plink1,plink

opal+https://opal-demo.obiba.org/ws/files/projects/RSRC/gps_data_final.Rdata

https://htsget.ga4gh.org/variants/1000genomes.phase1.chr1?format=VCF&referenceName=1&start=1&end=100000

https://ega.ebi.ac.uk:8052/elixir/tickets/tickets/EGAF00001753756?referenceName=chr21&start=1&end=100"

Association studies



Association studies (glm)

Meta-analysis

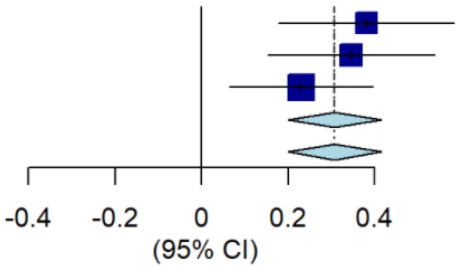
```
mod.meta <- ds.glmSLMA("DIS_DIAB ~ LAB_TRIG + GENDER",
                       dataName = "D" , family="binomial")
```

```
$output.summary$input.beta.matrix.for.SLMA
            betas study 1 betas study 2 betas study 3
(Intercept)    -5.1696619    -5.0254035    -4.4316966
LAB_TRIG        0.3813891     0.3462886     0.2304324
GENDER         -0.2260851    -0.4550068    -0.5416610

$output.summary$input.se.matrix.for.SLMA
            ses study 1 ses study 2 ses study 3
(Intercept)   0.4549299  0.39486781  0.29715719
LAB_TRIG      0.1037606  0.09880009  0.08471115
GENDER        0.4375805  0.39495629  0.29607590
```

Source	(95% CI)
1	0.38 [0.18; 0.58]
2	0.35 [0.15; 0.54]
3	0.23 [0.06; 0.40]
Total (fixed effect)	0.31 [0.20; 0.41]
Total (random effects)	0.31 [0.20; 0.41]

Heterogeneity: $\chi^2_2 = 1.49$ ($P = .47$), $I^2 = 0\%$



Pool analysis

```
mod <- ds.glm("DIS_DIAB ~ LAB_TRIG + GENDER", data = "D" , family="binomial")
mod$coeff
```

```
              Estimate Std. Error    z-value       p-value low0.95CI.LP high0.95CI.LP       P_OR
(Intercept) -4.7792110 0.21081170 -22.670521 8.755236e-114   -5.1923944   -4.36602770 0.00833261
LAB_TRIG     0.3035931 0.05487436   5.532514  3.156737e-08    0.1960414    0.41114488 1.35471774
GENDER      -0.4455989 0.20797931  -2.142516  3.215202e-02   -0.8532309   -0.03796695 0.64044060
            low0.95CI.P_OR high0.95CI.P_OR
(Intercept)    0.005527953      0.01254229
LAB_TRIG       1.216577226      1.50854390
GENDER         0.426036242      0.96274475
```

GWAS analysis in multi-cohort studies

GWAS Hypertension

Controlled access

- WTCCC
- GENOA
- ELSA

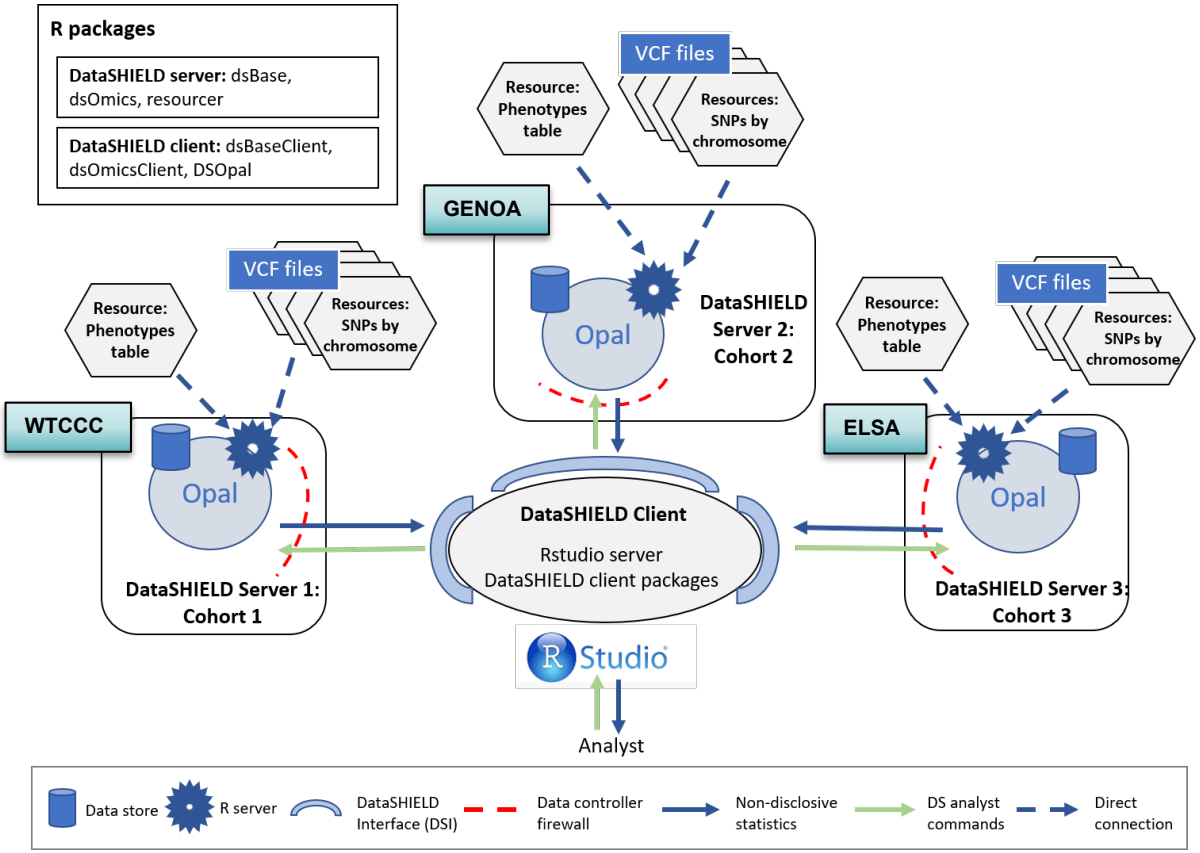
Data Analyst

- Allele frequency
- Single association
- Single GWAS
- Meta-analysis
- Mega-analysis

Bioconductor

plink... Whole genome association analysis toolset

PYTHO



GWAS analysis in multi-cohort studies

Load libraries

```
require('DSOpal')
require('dsBaseClient')
require('dsOmicsClient')
```

Logging server

```
builder <- DSI::newDSLoginBuilder()
builder$append(server = "cohort1", url = "https://opal-demo.o
               user =  "dsuser", password = "password",
               driver = "OpalDriver", profile = "omics")
builder$append(server = "cohort2", url = "https://opal-demo.o
               user =  "dsuser", password = "password",
               driver = "OpalDriver", profile = "omics")
builder$append(server = "cohort3", url = "https://opal-demo.o
               user =  "dsuser", password = "password",
               driver = "OpalDriver", profile = "omics")
logindata <- builder$build()
conns <- DSI::datashield.login(logins = logindata)
```

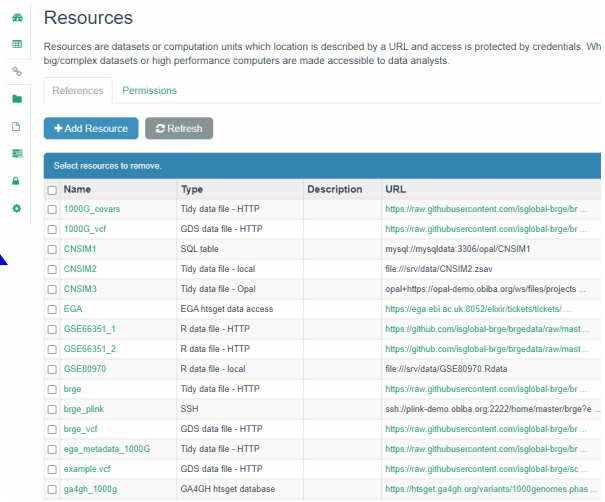
Load resources

```
  # Cohort 1 resources
  lapply(1:21, function(x){
    DSI::datashield.assign.resource(conns[1], paste0("chr", x),
    })

  # Cohort 2 resources
  lapply(1:21, function(x){
    DSI::datashield.assign.resource(conns[2], paste0("chr", x),
    })

  # Cohort 3 resources
  lapply(1:21, function(x){
    DSI::datashield.assign.resource(conns[3], paste0("chr", x),
    })
```

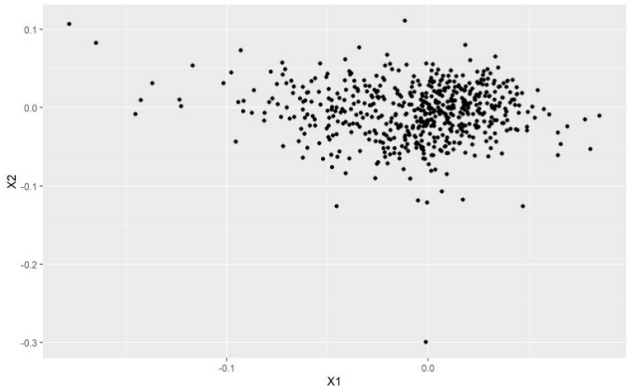
Opal



GWAS analysis in multi-cohort studies

PCA

```
pca <- ds.PCASNPS("ega_gds", prune = TRUE)
plotPCASNPS(pca)
```



Allele frequency

```
ds.alleleFrequency('geno')
```

```
$study1
# A tibble: 154 x 8
       M     F   all   n.M   n.F     n      MAF rs
   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>    <dbl> <chr>
 1 0.850 0.862 0.856   525   567  1092 0.144    rs58108140
 2 0.986 0.977 0.981   525   567  1092 0.0188   rs189107123
 3 0.890 0.882 0.886   525   567  1092 0.114    rs180734498
 4 0.976 0.970 0.973   525   567  1092 0.0270   rs144762171
 5 0.981 0.978 0.979   525   567  1092 0.0206   rs151276478
 6 0.253 0.295 0.275   525   567  1092 0.275    rs140337953
 7 0.992 0.991 0.992   525   567  1092 0.00824  rs187298206
 8 0.886 0.899 0.892   525   567  1092 0.108    rs116400033
 9 0.999 1     1.00    525   567  1092 0.000458 rs190452223
10 1     1     1       525   567  1092 0        rs181754315
# ... with 144 more rows
```

Single association analysis

```
ds.glmSNP(snps.fit = "rs58108140",
          model = casco ~  Gender + Population,
          genoData='geno')
```

```
              Estimate Std. Error   p-value    n     p.adj
rs58108140 0.003231591  0.1390788 0.9814623 1092 0.9814623
attr(,"class")
[1] "dsGlmSNP" "matrix"   "array"
```

And the same can be performed for a set of SNPs

```
ds.glmSNP(snps.fit = c("rs58108140", "rs189107123", "rs180734498"),
          model = casco ~ Gender + Population, genoData='geno')
```

```
               Estimate Std. Error   p-value    n     p.adj
rs58108140   0.003231591  0.1390788 0.9814623 1092 0.9814623
rs189107123 -0.301171963  0.3312846 0.3632955 1092 0.9814623
rs180734498  0.063508407  0.1505633 0.6731671 1092 0.9814623
attr(,"class")
[1] "dsGlmSNP" "matrix"   "array"
```

GWAS analysis in multi-cohort studies

GWAS

```
ds.GWAS('geno', model = casco ~ Gender + Population)
```

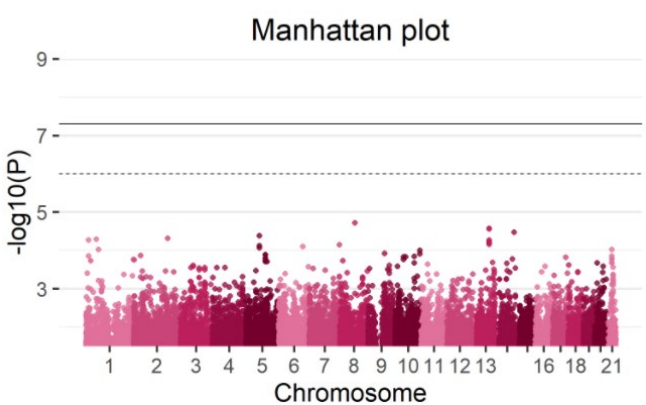
Cohort 1 Cohort 2 Cohort 3

```
# A tibble: 651,700 x 10
   rs          chr        pos n.obs  freq  p.value    Est Est
 * <chr>       <chr>    <int> <dbl> <dbl>    <dbl>  <dbl>  <d
 1 rs79460780  1       5.46e7   416 0.862  5.09e-5 -0.777  0.
 2 rs41270277  1       1.60e7   416 0.903  5.38e-5  0.910  0.
 3 rs12130802  1       6.63e7   418 0.920  9.41e-5  1.04   0.
 4 rs74054799  1       1.59e7   418 0.907  1.41e-4  0.868  0.
 5 rs12130716  1       2.48e8   417 0.620  1.73e-4 -0.541  0.
 6 rs78099457  1       2.66e7   416 0.808  1.85e-4 -0.665  0.
```

Cohort 1 Cohort 2 Cohort 3

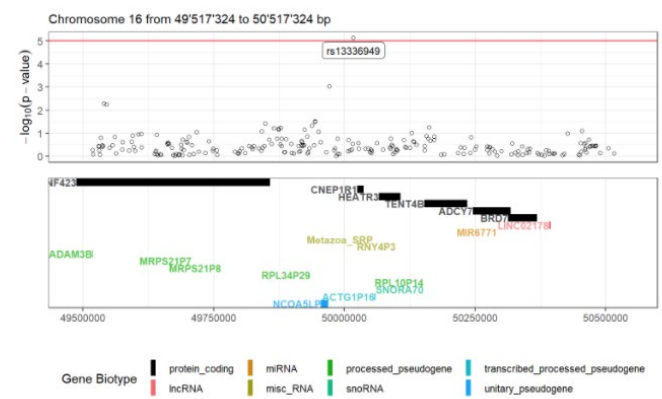
Manhattan plot

```
manhattan(results[[1]])
```



Locus Zoom

```
LocusZoom(meta_pvalues, pvalue = "p.meta")
```



GWAS analysis in multi-cohort studies

Table 4: **Beta values, standard errors and p-values yielded by the three methods: GWAS of all individuals, meta-analysis of synthetic cohorts, pooled analysis of synthetic cohorts.**

	Beta			SE			P-Value		
SNP id	Original	Pooled	Meta	Original	Pooled	Meta	Original	Pooled	Meta
rs72644130	0.502	0.490	0.500	0.119	0.125	0.122	2.5×10^{-5}	8.4×10^{-5}	3.9×10^{-5}
rs12408667	0.397	0.393	0.405	0.098	0.101	0.099	5.0×10^{-5}	9.2×10^{-5}	4.4×10^{-5}
rs28611360	0.331	0.319	0.344	0.084	0.086	0.086	7.5×10^{-5}	2.0×10^{-4}	6.0×10^{-5}
rs2377999	0.331	0.318	0.344	0.084	0.086	0.086	7.8×10^{-5}	2.1×10^{-4}	6.2×10^{-5}
rs12106789	−0.320	−0.316	−0.325	0.078	0.080	0.080	4.2×10^{-5}	8	
rs17713681	0.292	0.316	0.347	0.077	0.079	0.080	1.5×10^{-4}	6	
rs7644029	0.329	0.332	0.370	0.088	0.091	0.091	2.0×10^{-4}	2	
rs13236153	0.281	0.278	0.304	0.074	0.076	0.076	1.6×10^{-4}	2	
rs569538672	−0.470	−0.481	−0.485	0.116	0.120	0.117	5.0×10^{-5}	6	

	MSE	Bias
Beta		
Pooled	2.8×10^{-4}	4.5×10^{-3}
Meta	1.1×10^{-3}	-1.0×10^{-2}

Examples

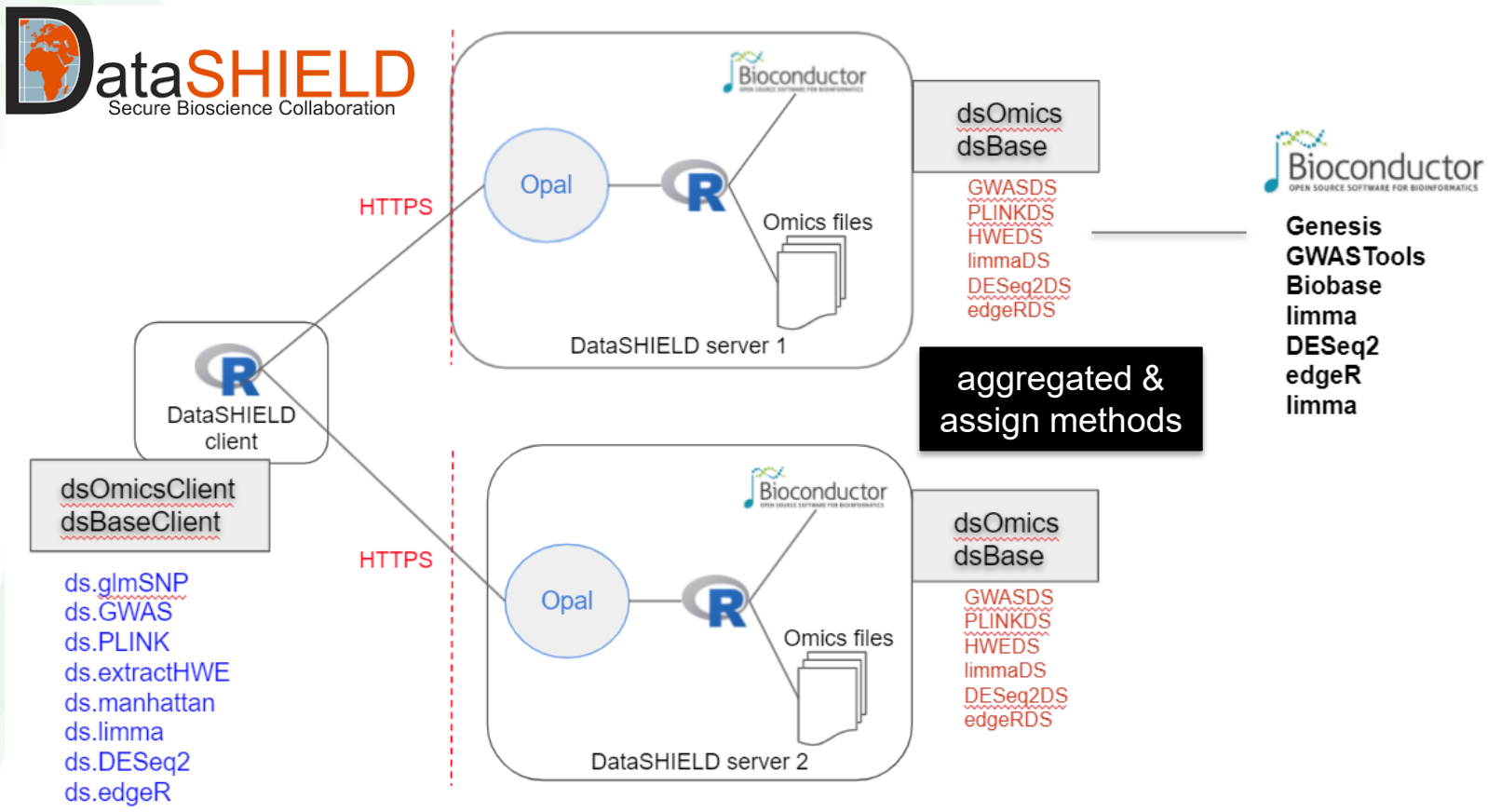
- **LifeCycle: M**arkers of early-life stressors that influence health across the lifecycle (>25 cohorts)

- **AHTLETE:** Birth cohorts studying the role of the exposome in health (epigenome vs greenspace, exposome trajectories and health, …)

- **Estonian Biobank:** 50,000 genomes and medical records (association analyses, …)

- **Others:** EUCAN connect, CINECA, EGA, …

How DataSHIELD packages are created



Discussion

General questions?

Important issues

- **GPDR compliance**
  - Original data cannot be access from client side
  - The “profiles” in Opal: level of analysis, every cohort control their own rights, sets of functions, use of packages
  - Non-disclosive analyses
  - Control filter low number

- **Reproducibility:** Data can be stored and managed by the owner and provide access trough to DS to the reviewers and other researchers

- **Risk assessment:** Each proposal is assessed against the risk of having disclosive analyses.

Characteristics

- Build on robust hardware and encrypt all transmissions
- Build upon formal governance agreements
- Serverside R only callable via Opal
- Stored data in Opal servers are pseudonymized (i.e. remove direct identifiers such as real name, real ID, etc.)
- Parser – allows only valid characters/functions
- Serverside functions don't allow disclosive output (e.g. print; glm: residuals & fitted values, etc)
- Disclosure traps (e.g. min cell size; glm saturation)
- Data custodian controls trap-thresholds
- Log all commands/output on remote servers
- Attach DataSHIELD to a bespoke database