

# Science-DB

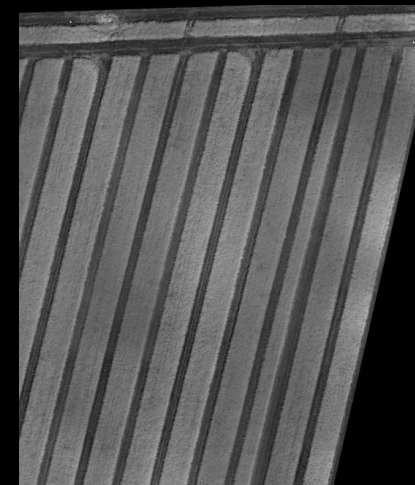
A knowledge and data exchange  
platform for science

# Typical problems

marker	mother_penn_nimbus	father_penn	father_nimbus	mother_hattrick	father_hattrick
Bn-A01-p1001022	G	G	G	G	G
Bn-A01-p10026020	G	G	A	R	A
Bn-A01-p10027328	A	A	G	R	G
Bn-A01-p100441	T	T	T	T	T
Bn-A01-p100693	T	T	T	T	T
Bn-A01-p10105499	T	T	C	Y	C

race	cultivar	level	multiplier	batch	batch_status	storage	harvest_year	booking_date	admission_number	laboratory	examination_date	purity
3955 RAW	RAW	BS	7000347	MR 1404	TMTD gebeizt	3	2013	2013-08-01 00:00:00.000	DE013-1472005	NPZ	2013-08-03 00:00:00.000	0
3955 RAW	RAW	BS	7000347	MR 1404	TMTD gebeizt	3	2013	2013-08-01 00:00:00.000	DE013-1472005	NPZ	2013-08-03 00:00:00.000	0
3955 RAW	RAW	BS	7000347	MR 1404	TMTD gebeizt	3	2013	2013-08-01 00:00:00.000	DE013-1472005	NPZ	2013-08-05 00:00:00.000	99.8
3955 RAW	RAW	BS	7000347	MR 1404	TMTD gebeizt	3	2013	2013-08-01 00:00:00.000	DE013-1472005	NPZ	2013-08-06 00:00:00.000	0

Marker	chr	position
Bn_A01_p1000115	chrA01	617757
Bn_A01_p100441	chrA01	2468965
Bn_A01_p100693	chrA01	2469216
Bn_A01_p101021	chrA01	2469547
Bn_A01_p101094	chrA01	2469620
Bn_A01_p101451	chrA01	2469977
Bn_A01_p10181972	chrA01	8782944
Bn_A01_p101943	chrA01	2470471
Bn_A01_p10231780	chrA01	8828045
Bn_A01_p10249225	chrA01	8852208
Bn_A01_p10250255	chrA01	8853237
Bn_A01_p10251052	chrA01	8854034



# Knowledge exchange platform

- Save store for project data
- Easy interchange of data
  - Big data made accessible
- (Mathematically) well defined data models
  - Relations between data
- Public / private repository
  - Link to published (e.g. NCBI, EBI) or other closed databases possible

# Knowledge exchange platform

- Wiki (Documentation)
- Seafile (File storage)
- **Science-Db**
  - RESTful Api for data-management
    - Model View Controller (MVC) Architecture
    - CRUD cases
      - Bulk create (Excel) & Search
  - Web based graphical user interface (GUI)
  - Code generation

# Technical Requirements

- RESTful Web Services to enable easy access
- CRUD use cases (Create, Read, Update, and Delete; also Search and Ex-/Import)
- HTTP Api
- Quick to integrate new data definitions
- Random access to very large tables (e.g. SNP data)

# Database API

- Node JS (Javascript)
- Model View Controller (MVC) Architecture
- No, or minimal, coding required to generate new data models:
  - SQL tables
  - NodeJS model
  - NodeJS controller
  - VueJS based Web-GUI

# Example: *Plant* Data-Model

- Create new data-model definition:
  1. `sequelize model:create --name Plant --attributes 'title:string, taxon:string, gmo:boolean'`
  2. `sequelize db:migrate`
  3. `express_route_gen . --name Plant --attributes 'title:string, taxon:string, gmo:boolean'`

# Example: *Plant* Data Model

- Create new Plant:

- ✓ `curl -data "title=Plant_1&taxon=x&gmo=false" http://www.science.db/plants`

- Read Plants:

- ✓ **All:** `curl 'http://www.science.db/plants'`
  - ✓ **Paginate:** `curl 'http://www.science.db/plants?page=1&per_page=10'`
  - ✓ **Search:** `curl 'http://www.science.db/plants?sort=taxon|asc&page=1&per_page=20&filter=e'`



# Example: *Plant* Data Model

- Update existing Plant:

- ✓ `curl -X PUT --data "taxon=y"`  
`http://www.science.db/plant/1`

- Delete existing Plant:

- ✓ `curl -X DELETE`  
`http://www.science.db/plant/1`

# Excel / CSV Interface

- Download example CSV table:

```
- curl  
  'http://www.science.db/plants/example_csv'
```

- Upload populated CSV table:

```
- curl -F "csv_file=@plants_csv.csv"  
  www.science.db/plants/upload_csv
```

# Graphical User Interface (GUI)

- GUI code generated

Search for:

Go

Reset

#	<input type="checkbox"/>	Name ^	Shits	Fur_color	Actions
1	<input type="checkbox"/>	Alien	false	Transparant	<input type="button" value="🔍"/> <input type="button" value="✎"/> <input type="button" value="✕"/>
2	<input type="checkbox"/>	Ghost	false	White	<input type="button" value="🔍"/> <input type="button" value="✎"/> <input type="button" value="✕"/>
					<input type="button" value="🔍"/> <input type="button" value="✎"/> <input type="button" value="✕"/>

# Genome Browser (D3)



# Extremely large tables

- SNP data easily has 10k rows and millions of columns
- Fast random access is required
- Preferred RESTful HTTP-Frontend
- RASDAMAN (RESTful API and Web based GUI)

# Security

- Authorization of registered Users
- Authentication of actions through roles
  - Guest
  - Scientist
  - Administrator

# Example usage in R

```
# Load dependencies
require(httr)

# Login
login.res <- POST('http://213.136.88.239:3030/login',
body=list('email'='m.enders@npz-innovation.de',
'password'='####')
)

# Load all marker data
markers.res <- GET('http://213.136.88.239:3030/markers')

# To obtain the content of the response as a list of
lists, use:
markers.lst <- content(markers.res)
```

# Example usage in R

```
# Convert to data.frame (table)  
Reduce(rbind, lapply( content(markers.res) [1:3],  
as.data.frame,  
stringsAsFactors=FALSE )) [,1:4]
```

```
# Gives the following tabular output:  
id name chromosome position  
1 x chr1 12  
4 Bn_A01_p1000115 chrA01 617757  
5 Bn_A01_p100441 chrA01 2468965
```



# Science-DB

- Small, flexible, and robust API
  - Facilitates knowledge exchange in diverse research groups
- Fast and easy data model definition and extension, **minimal coding required**
  - Biological or any other data type
- Automated change history and backups
- Security

# Science-DB

Two access layers:

- Enables direct integration in computational analyses: R, Julia, Python, etc.
  - Big data management solved
- Comes with an intuitive Web Frontend