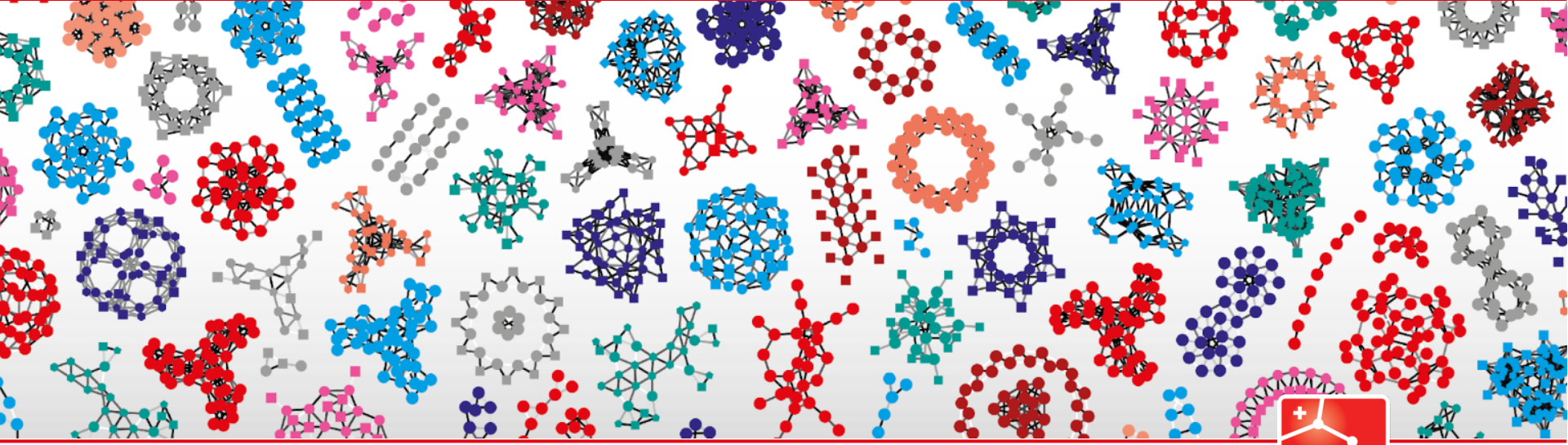


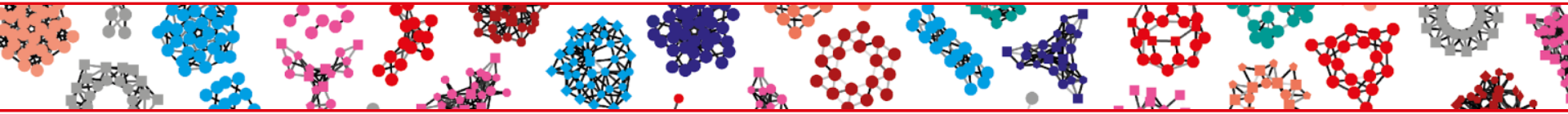
Swiss Institute of Bioinformatics

SIB a FAIR connected institute

Jerven Bolleman
SIB Employees and Members

elixir SWITZERLAND

Overview



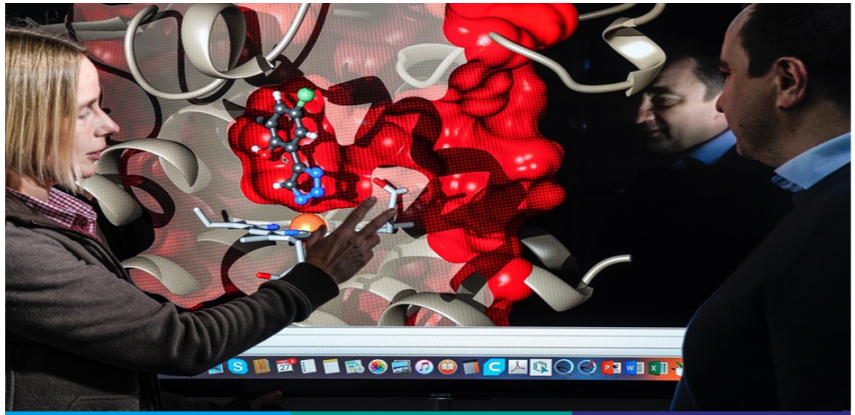
01 SIB

02 RDF

03 SPARQL

04 Federation

The five pillars of SIB's activities



Databases & Software Tools

Core Facilities & Competence Centres

Personalized Health

01 Infrastructure

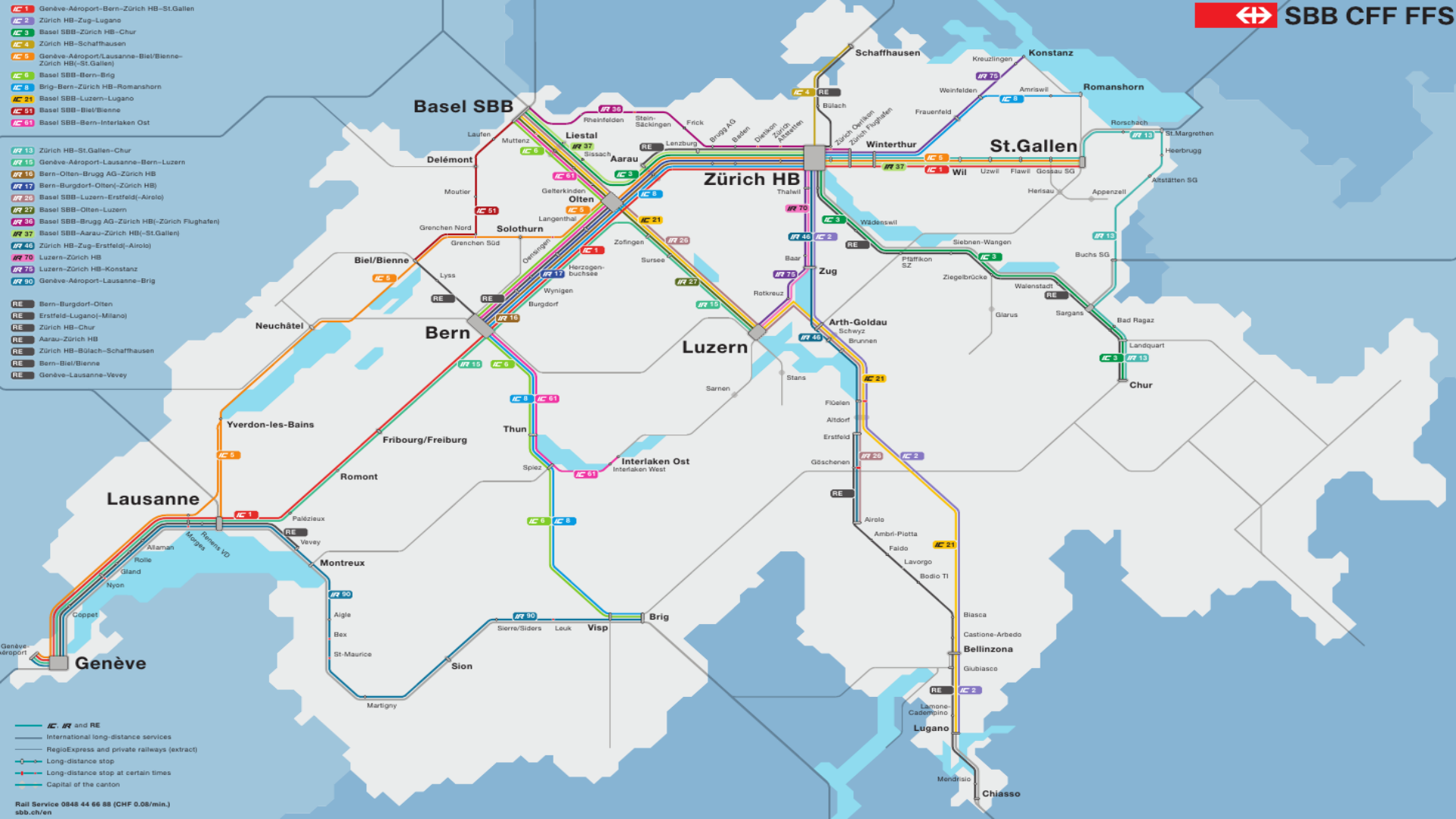
Scientific Collaboration

Training

02 Community

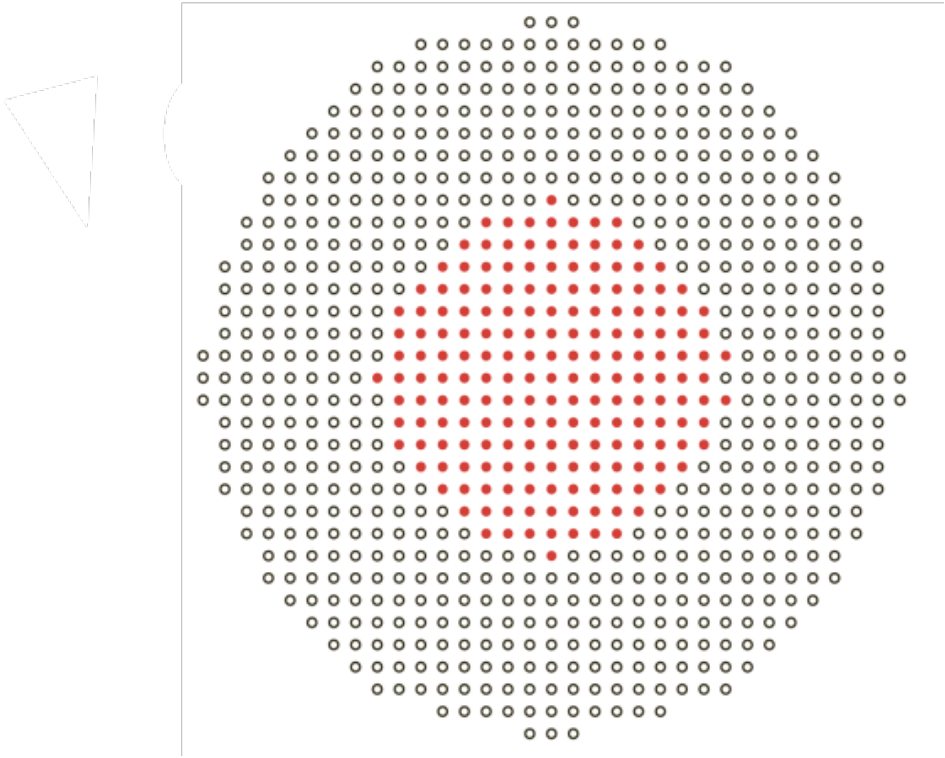
(Con)Federation

20 institutional partners all over Switzerland


SBB CFF FFS
IC 1 Genève-Aéroport–Bern–Zürich HB–St.Gallen
IC 2 Zürich HB–Zug–Lugano
IC 3 Basel SBB–Zürich HB–Chur
IC 4 Zürich HB–Schaffhausen
IC 5 Genève-Aéroport/Lausanne–Biel/Bienne–Zürich HB(–St.Gallen)
IC 6 Basel SBB–Bern–Brig
IC 8 Brig–Bern–Zürich HB–Romanshorn
IC 21 Basel SBB–Luzern–Lugano
IC 51 Basel SBB–Biel/Bienne
IC 61 Basel SBB–Bern–Interlaken Ost
IR 13 Zürich HB–St.Gallen–Chur
IR 15 Genève-Aéroport–Lausanne–Bern–Luzern
IR 16 Bern–Olten–Brugg AG–Zürich HB
IR 17 Bern–Burgdorf–Olten(–Zürich HB)
IR 26 Basel SBB–Luzern–Erstfeld(–Airolo)
IR 27 Basel SBB–Olten–Luzern
IR 36 Basel SBB–Brugg AG–Zürich HB(–Zürich Flughafen)
IR 37 Basel SBB–Aarau–Zürich HB(–St.Gallen)
IR 46 Zürich HB–Zug–Erstfeld(–Airolo)
IR 70 Luzern–Zürich HB
IR 75 Luzern–Zürich HB–Konstanz
IR 90 Genève-Aéroport–Lausanne–Brig
RE Bern–Burgdorf–Olten
RE Erstfeld–Lugano(–Milano)
RE Zürich HB–Chur
RE Aarau–Zürich HB
RE Zürich HB–Bülach–Schaffhausen
RE Bern–Biel/Bienne
RE Genève–Lausanne–Vevey
Basel SBB
Delémont
Moutier
Biel/Bienne
Lyss
Neuchâtel
Yverdon-les-Bains
Lausanne
Palézieux
Vevey
Montreux
Aigle
Bex
St-Maurice
Martigny
Sion
Sierre/Siders
Leuk
Visp
Brig
Genève
Coppet
Nyon
Rolle
Gland
Allaman
Morges
Renens VD
Fribourg/Freiburg
Romont
Bern
Grenchen Nord
Grenchen Süd
Solothurn
Olten
Gelterkinden
Langenthal
Herzogenbuchsee
Wynigen
Burgdorf
Langnau
Thun
Spiez
Interlaken Ost
Interlaken West
Liestal
Laufen
Muttenz
Rheinfelden
Stein-Säckingen
Frick
Sissach
Aarau
Lenzburg
Brugg AG
Baden
Dietikon
Zürich Altstetten
Zürich Oerlikon
Zürich Flughafen
Zofingen
Sursee
Luzern
Sarnen
Stans
Arth-Goldau
Schwyz
Brunnen
Rotkreuz
Zug
Baar
Thalwil
Wädenswil
Pfäffikon SZ
Ziegelbrücke
Siebnen-Wangen
Glarus
Walenstadt
Sargans
Bad Ragaz
Landquart
Chur
Flüelen
Altdorf
Erstfeld
Göschenen
Airolo
Ambri-Piotta
Faido
Lavorgo
Bodio TI
Biasca
Castione-Arbedo
Bellinzona
Giubiasco
Lamone-Cadempino
Lugano
Mendrisio
Chiasso
Winterthur
Schaffhausen
Bülach
Frauenfeld
Weinfelden
Kreuzlingen
Konstanz
Amriswil
Romanshorn
St.Gallen
Wil
Uzwil
Flawil
Gossau SG
Herisau
Appenzell
Rorschach
St.Margrethen
Heerbrugg
Altstätten SG
Buchs SG
IC, IR and RE
International long-distance services
RegioExpress and private railways (extract)
Long-distance stop
Long-distance stop at certain times
Capital of the canton
Rail Service 0848 44 66 88 (CHF 0.08/min.)
sbb.ch/en

A national community of bioinformaticians

SIB has 620 affiliated members
and 190 employees

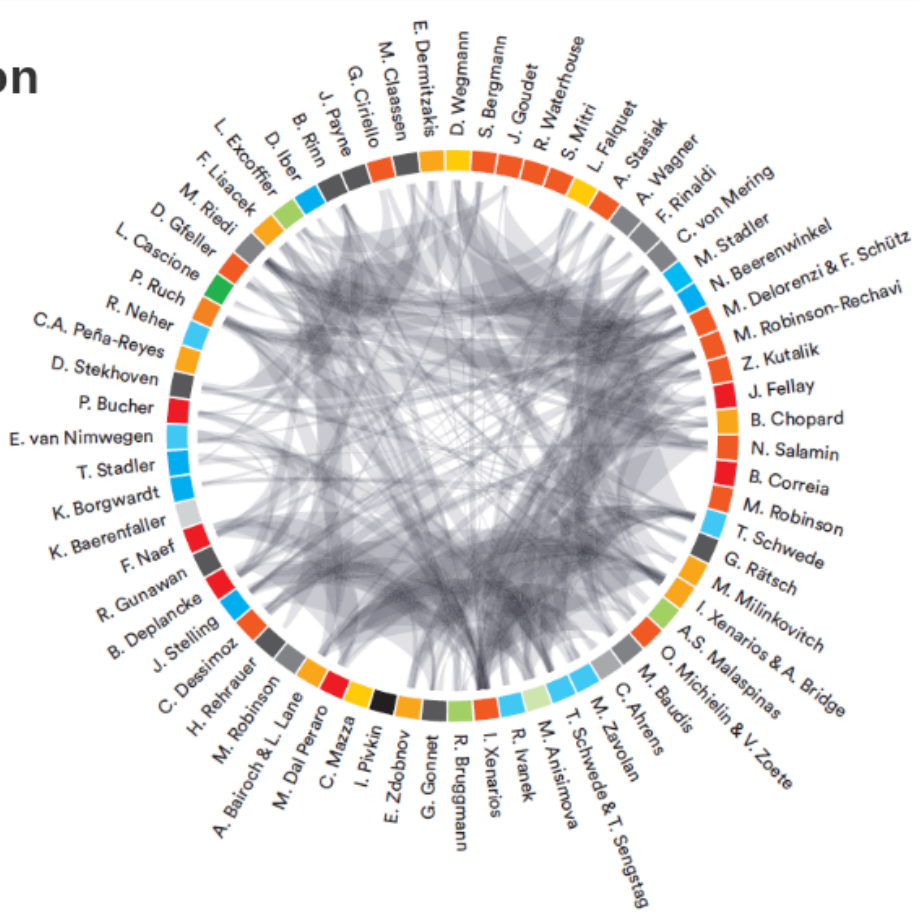


Fostering scientific collaboration

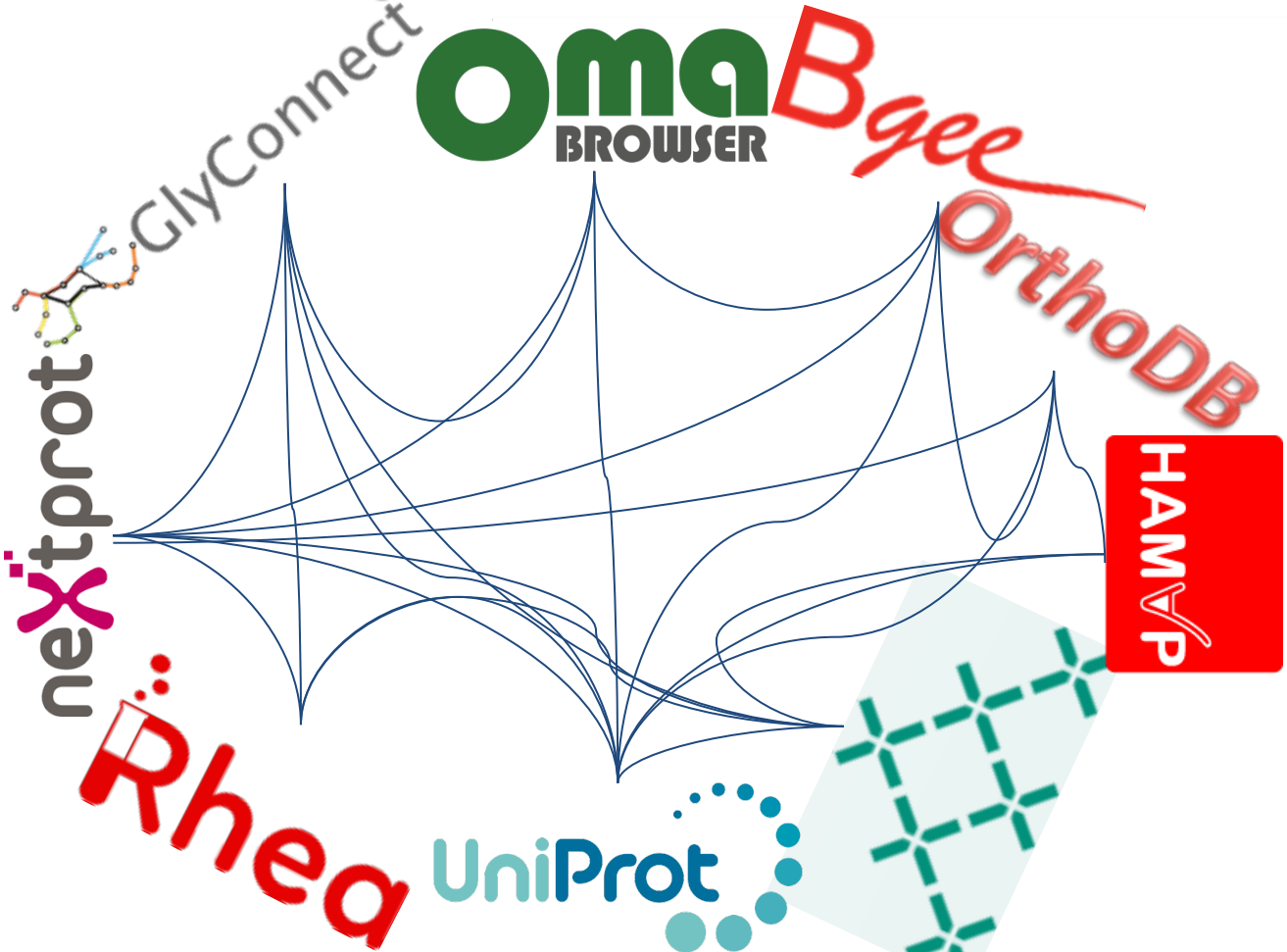
SIB collaboration network

Past or ongoing collaborations between SIB groups

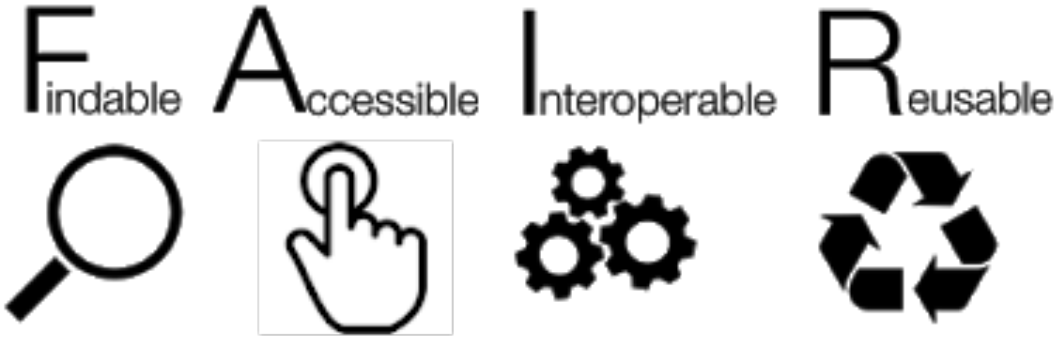
254 publications in 2018



Fostering scientific collaboration

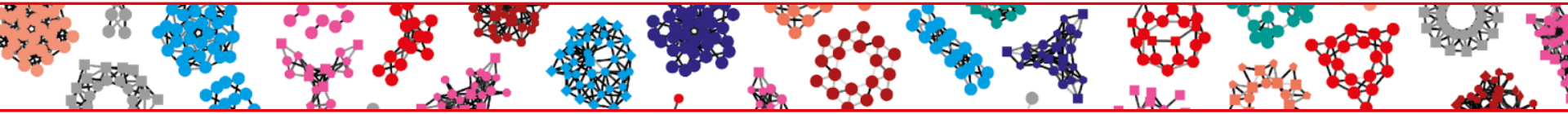


The FAIRest format of them all



Hell is other people’s data

Overview



01 SIB

02 RDF

03 SPARQL

04 Federation

（聯合的蛋白）是一个全面的，高质量的，免费使用的蛋白质序列与功能信息数据库，许多内容来自基因组计划，它还包含了大量来自研究文献的关于蛋白的生物学功能信息

يونيبروت بالإنجليزية هي قاعدة بيانات شاملة ذات جودة عالية مجانية لتسلسل البروتين والمعلومات الوظيفية، استمدت الكثير من المدخلات من مشاريع الجينوم.

تحتوي قاعدة البيانات على كمية كبيرة من المعلومات المستخلصة من

Resource Description Framework

- **Statement**
  Also known as triples, subject → predicate → object

- **IRI**
  Also known as URI often an URL (IRI allows 唄, URL locates and identifies)

- **Literal**
  “Strings”, “chaîne”@fr, true, 1, 1.000001
  “2001”^^xsd:gYear, “2001-10-11”^^xsd:date, “2001-10-11T09:30:10”^^xsd:dateTime

- **Bnode**
  Placeholder identifiers (something exists but you don’t know what)

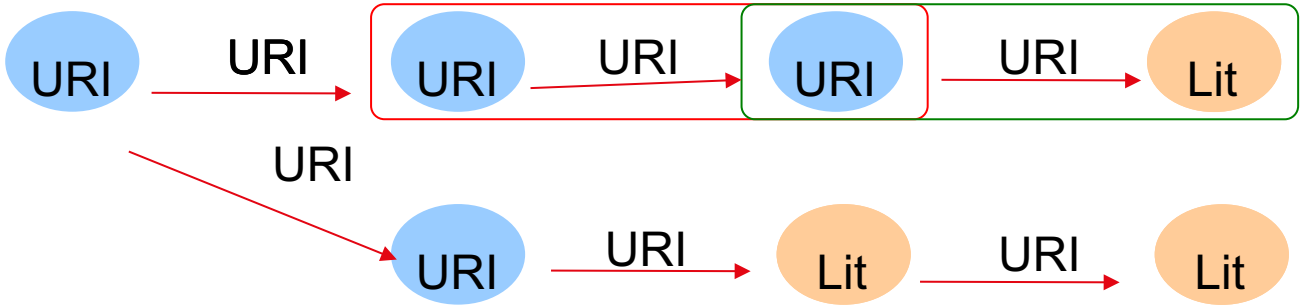
RDF statements: from triples to graphs

Triples:

Subject 1 + Predicate 1 + Object 1

Subject 2 + Predicate 2 + Object 2

Graph:



RDF statements: from triples to graphs

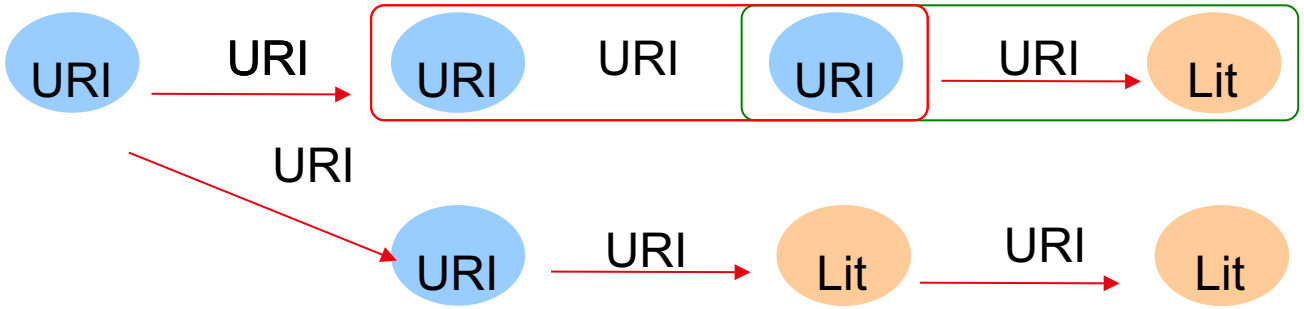
Triples:

Subject 1 + Predicate 1 + Obje

Subject 2 + Predicate 2 + Obje

Global identifiers – semantic standards – make data integration on the fly possible.

Graph:



Example (format ntriples)

<http://purl.uniprot.org/uniprot/P05067> <http://purl.uniprot.org/core/reviewed> true .

<http://purl.uniprot.org/uniprot/P05067> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> ↷
<http://purl.uniprot.org/core/Protein> .

<http://purl.uniprot.org/uniprot/P05067> <http://purl.uniprot.org/core/created> ↷
"1987-08-13"^^ <http://www.w3.org/2001/XMLSchema#date> .

<http://purl.uniprot.org/uniprot/P05067> <http://purl.uniprot.org/core/created> ↷
_:1

Example (turtle)

```
base <http://purl.uniprot.org/uniprot/>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix up: <http://purl.uniprot.org/core/>
prefix xsd: <http://www.w3.org/2001/XMLSchema#>

<P05067> rdf:type up:Protein ;
  up:reviewed true ;
  up:created "1987-08-13"^^xsd:date ;
```

Other formats

- RDF/XML
  - Oldest standard
- JSON-LD
  - Can be used to map from JSON
- RDFa
  - For embedding in HTML
- HDT
  - Compressed but still queryable

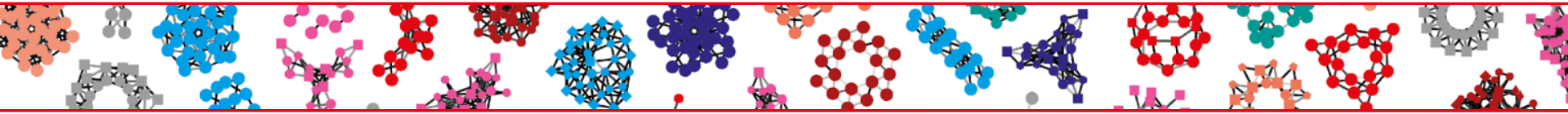
SIB RDF

- **Since 2004**
  - **With UniProt**
  - **Bottom up take up by other resources**
- **Linking is key !**
  - **200+ Cross-References to other database**
- **Identify all the things**

Your RDF

- **Key**
  - **IRI per resource**
  - **One thing one identifier**
- **Format not so important**
- **Reuse ontologies,**
  - **after understanding your own data**

Overview



01 SIB

02 RDF

03 SPARQL

04 Federation

Lots of implementations

MERCK

Virtuoso

Openlinksw

AllegroGraph

SPARQL basics

- Remember Turtle?
  - Since 1.1 looks like SPARQL

```
1 SELECT
2   *
3 WHERE
4 {
5   ?s ?p ?o .
6 }
```

Take our turtle example

```
1 base <http://purl.uniprot.org/uniprot/>
2 prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3 prefix up: <http://purl.uniprot.org/core/>
4 prefix xsd: <http://www.w3.org/2001/XMLSchema#>
5
6 <P05067> rdf:type up:Protein ;
7   up:reviewed true ;
8   up:created "1987-08-13"^^xsd:date ;
9
```

Introduce a variable

```
1 base <http://purl.uniprot.org/uniprot/>
2 prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3 prefix up: <http://purl.uniprot.org/core/>
4 prefix xsd: <http://www.w3.org/2001/XMLSchema#>
5
6 ?protein rdf:type up:Protein ;
7   up:reviewed true ;
8   up:created "1987-08-13"^^xsd:date ;
9
```

Add a SELECT and a WHERE

```
1  PREFIX up:<http://purl.uniprot.org/core/>
2  PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3  PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>
4
5  SELECT
6    ?protein
7  WHERE {
8    ?protein rdf:type up:Protein ;
9      up:reviewed true ;
10     up:created "1987-08-13"^^xsd:date .
11 }
```

Four query kinds

- SELECT
  - Gives a TABLE
- CONSTRUCT
  - Gives more RDF
- ASK
  - True or False

- DESCRIBE
  - Some RDF about the node you asked for
  - Not used in programs
  - For data discovery as human only

ASK

```
PREFIX up:<http://purl.uniprot.org/core/>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>

ASK {
  ?protein rdf:type up:Protein ;
    up:reviewed true ;
    up:created "1987-08-13"^^xsd:date .
}
```

CONSTRUCT

```
1  PREFIX up:<http://purl.uniprot.org/core/>
2  PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3  PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>
4
5  CONSTRUCT {
6    ?protein rdf:type <http://example.org/MY_FAV>
7  }
8  WHERE
9  {
10   ?protein rdf:type up:Protein ;
11     up:reviewed true ;
12     up:created "1987-08-13"^^xsd:date .
13 }
```

DESCRIBE

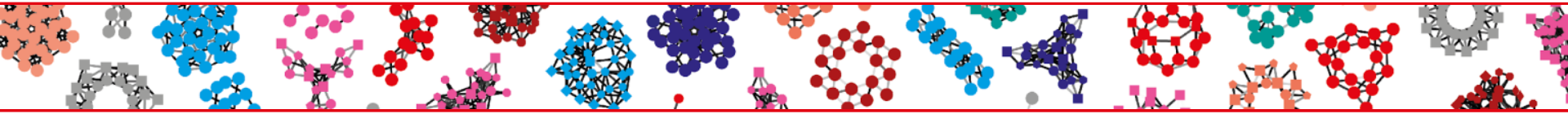
Only to discover datasets

```
1 BASE <http://purl.uniprot.org/uniprot/>
2
3
4 DESCRIBE <P05067>
```

Download formats

- SELECT + ASK
  - application/sparql-results+json
  - application/sparql-results+xml
  - Often
    - CSV
    - TSV
    - HTML
- Construct + Describe
  - Ntriples (nt)
  - RDF/XML (rdf)
  - Turtle (ttl)
  - Often
    - RDFa

Federation



01 SIB

02 RDF

03 SPARQL

04 Federation

Ask DB 2 to be part of the solution

```
1  PREFIX up:<http://purl.uniprot.org/core/>
2  PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3  PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>
4
5  SELECT
6    ?protein ?other
7  WHERE {
8    ?protein rdf:type up:Protein ;
9      up:reviewed true ;
10     up:created "1987-08-13"^^xsd:date .
11   SERVICE <https://www.ebi.ac.uk/rdf/services/sparql>{
12     ?other ?x ?protein
13   }
14 }
```

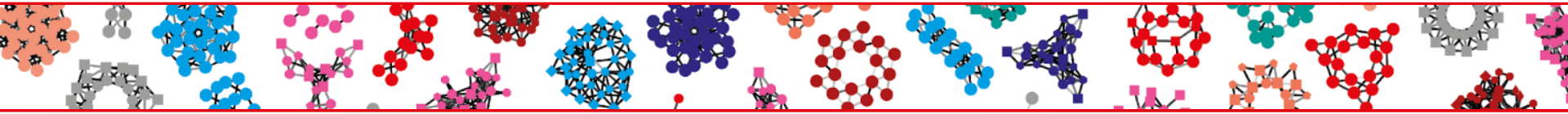
Ask DB 2 to be part of the solution

```
PREFIX up:<http://purl.uniprot.org/core/>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>

SELECT
  ?protein ?other
WHERE {
  ?protein rdf:type up:Protein ;
    up:reviewed true ;
    up:created "1987-08-13"^^xsd:date .
  SERVICE <https://www.ebi.ac.uk/rdf/services/sparql>{
    SELECT
        ?other
    WHERE {
        ?other ?x ?protein
    } LIMIT 10
  }
}
```

Issue when federating

- Bugs add up
  - SPARQL 1.1 compliance
  - Downtime
- Optimizers are blind
  - Order
  - Distance
  - Endpoints change
    - Number and kind of triples
    - Implementation



Thank you for part one