

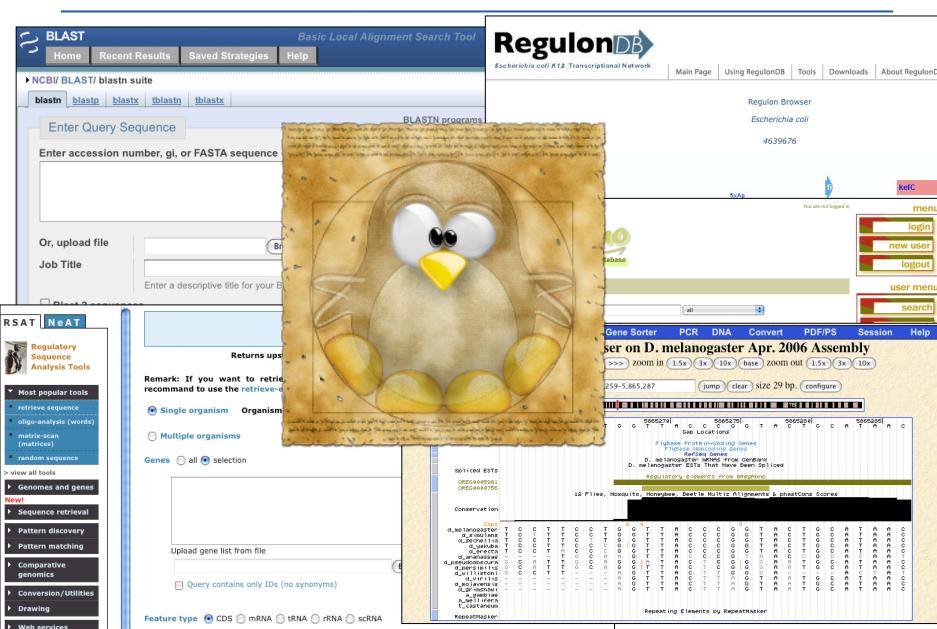


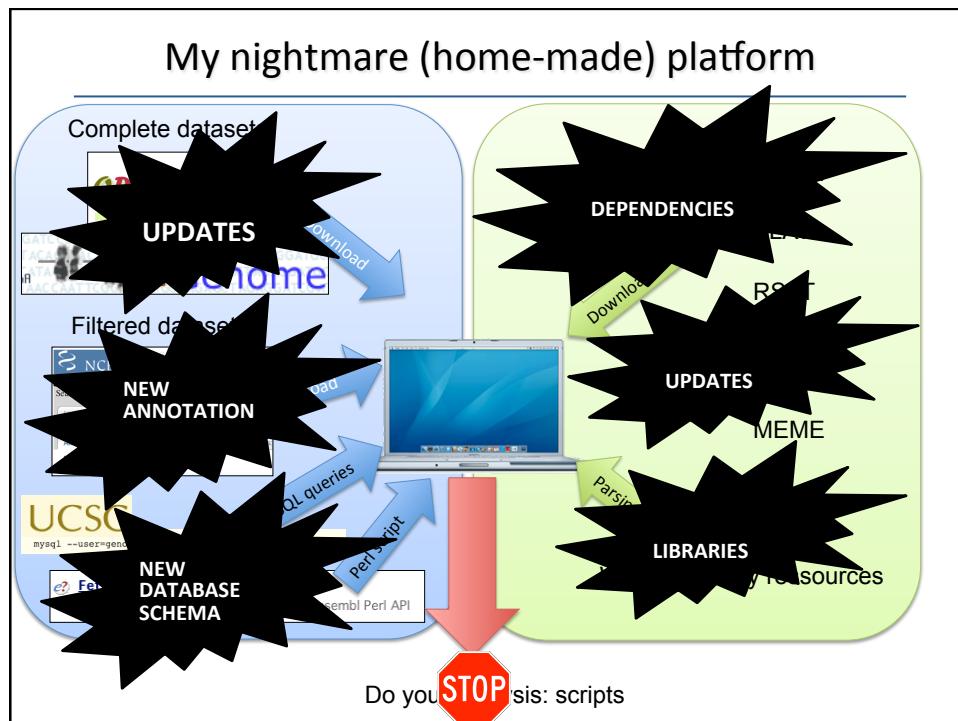
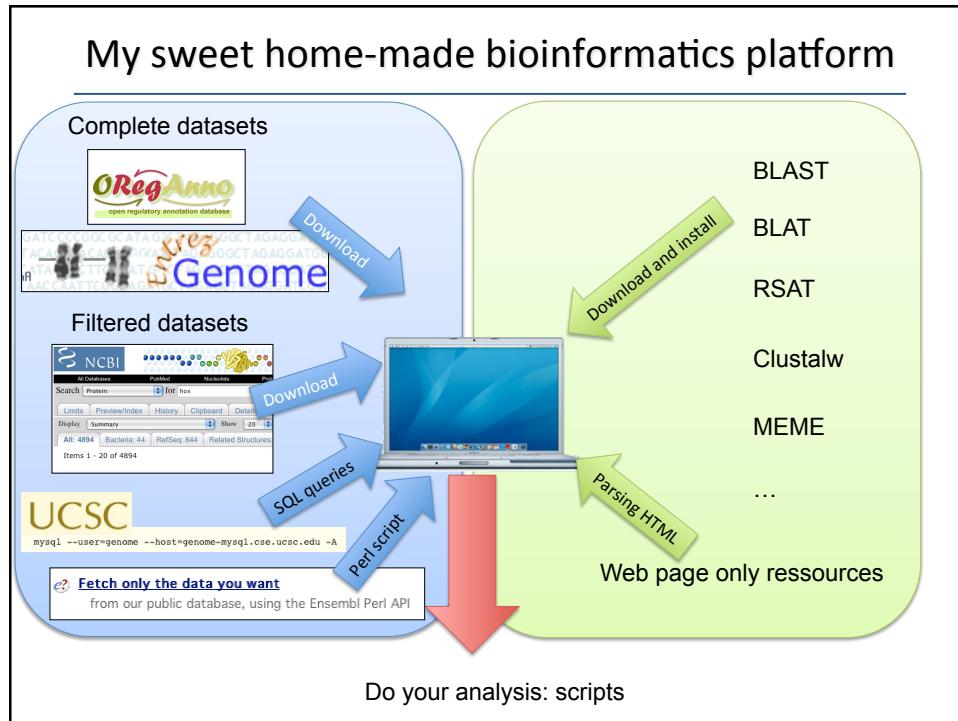
## TUTORIAL T01 ECCB 2014

### Analysis of Cis-Regulatory Motifs from High-Throughput Sequence Sets

#### Session 3 : SOAP Web services

A bioinformatics « world » for humans





**Welcome to the genomic era !**

---

**Find a Species**

**Ensembl Species**

	Aedes
	Alpaca
	Anole lizard (preview - assembly only)
	Anopheles
	Armadillo
	Bushbaby
	C. elegans
	C. intestinalis
	C. savignyi
	Cat
	Chicken
	Chimpanzee
	Cow
	Dog
	Dolphin
	Elephant
	Fruitfly
	Fugu
	Gorilla
	Guinea Pig
	Hedgehog
	Horse
	Human
	Hyrax
	Kangaroo rat
	Lamprey (preview - assembly only)
	Lesser hedgehog tenrec
	Macaque
	Medaka
	Megabat
	Microbat
	Mouse
	Mouse Lemur
	Opossum
	Orangutan
	Pika
	Platypus
	Rabbit
	Rat
	S. cerevisiae
	Shrew
	Squirrel
	Stickleback
	Tarsier
	Tetraodon
	Tree Shrew
	X. tropicalis
	Zebrafish

**Sequencing technologies — the next generation**

Michael L. Meckler

Never has the state of DNA sequencing technology been in greater flux than today. The resultant effect of the technology's rapid improvements is that it is becoming increasingly difficult to keep up with the latest developments. It is becoming increasingly difficult to keep up with the latest developments. This article summarizes some of the latest advances in sequencing technologies and their implications for greater volumes than ever before. These new generation sequencing technologies have the potential to revolutionize the way we approach the analysis of single plants or capable of addressing the needs of most research. The implications are far-reaching and far-reaching. With this in mind, the implications are far-reaching and far-reaching.

**1000 Genomes**  
A Deep Catalog of Human Genetic Variation

Home About Partners Data Contact Wiki

**1001 Genomes**  
A Catalog of *Arabidopsis thaliana* Genetic Variation

Home Collaborators Accessions Tools Downloads About Help desk

**The ENCODE Project: ENCyclopedia Of DNA Elements**

ENCODE

- Overview
- Consortium Membership
- Data Release Policy
- Expressions GENE Data
- Common Cell Types
- Requests for Application (RFAs)
- Press Releases and Publications
- Program Staff

**Genome Sequencing Projects on GOLD ©**  
Jan 2009, 4370 projects

Date	Incomplete Projects	Complete Projects	Total Projects
Dec-1997	0	0	0
Jul-1998	0	0	0
Jul-1999	0	0	0
Jul-2000	0	0	0
Jul-2001	0	0	0
Jul-2002	0	0	0
Jul-2003	0	0	0
Aug-2003	0	0	0
Feb-2004	0	0	0
Jun-2004	0	0	0
Oct-2004	0	0	0
Oct-2005	0	0	0
Apr-2006	0	0	0
Oct-2006	0	0	0
Mar-2007	0	0	0
May-2008	0	0	0
Jan-2009	0	0	0
May-2009	4370	0	4370

**nature REVIEWS GENETICS**  
**nature genetics**

**Bye bye home-made platform...**

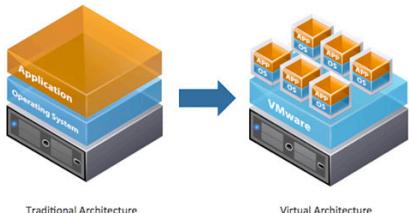
---

<http://www.genomequest.com/landing-pages/ODI-webinar-web.html>

## Virtual machines

---

**Virtualization Defined**  
For those more visually inclined...



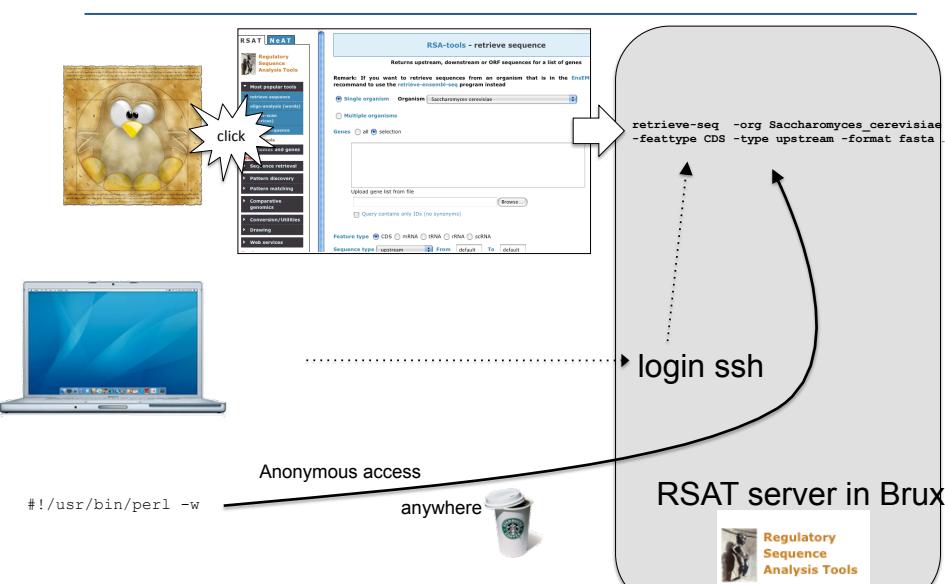
Traditional Architecture      Virtual Architecture

**All-in-a-box option**  
 ⇒ Large files  
 ⇒ Does not fix the issue of data download



Source : vmware.com

## Programs « talking » to programs



click

Anonymous access anywhere

```
#!/usr/bin/perl -w
```

retrieve-seq -org Saccharomyces\_cerevisiae -feattype CDS -type upstream -format fasta

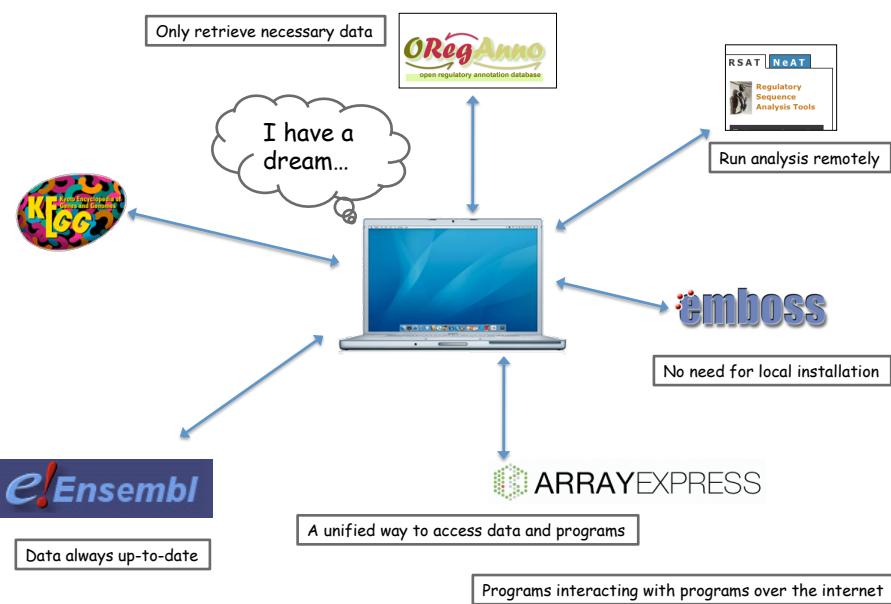
login ssh

RSAT server in Bruxelles

## A future bioinformatics « world » for computers ?



## A future bioinformatics « world » for computers ?



## Web Services to the rescue ?

**commentary**

### Creating a bioinformatics nation

A web-services model will allow biological data to be fully exploited.

**Lincoln Stein**

During the Middle Ages and early Renaissance, Italy was fragmented into dozens of rival city-states controlled by such legendary families as the Estes, Viscontis and Medicis. Though picturesque, this political fragmentation was ultimately damaging to science and commerce because of the lack of standardization in everything from weights and measures to the tax code to the currency to the very dialects people spoke. A fragmented and technologically weak society was vulnerable to conquest, and from the seventeenth to the nineteenth centuries Italy was dominated by invading powers.

The old city-states of Italy are an apt metaphor for bioinformatics today. The field is dominated by rival groups, each promoting its web sites, services and data formats independently; the resulting competitive chaos has greatly enriched the field. But it has also created a significant hindrance to researchers wishing to exploit the wealth of genome data to its fullest.

Despite its shaky beginning, the nation of Italy was eventually forged through a combination of violent and diplomatic efforts. It is

Figure 1 Moving towards a bioinformatics nation. Because each data provider (such as UCSC) publishes data in an idiosyncratic form, the Bio\* software package (Bio\*) attempts to massage data into a standard internal format. Unfortunately, Bio\* needs to be fixed, provider changes its formats. A web-services world would build on the successes of the defining standard interfaces to various types of computations and data formats. The Bio\* needs to be written to recognize these interfaces, allowing them to interoperate easily with all data. A service registry would let data providers enter an electronic 'address book', allowing Bio\* libraries to locate and interact with new data sources automatically.

« Although this proposal may seem a far cry from what happens now, the technology exists to make it reality. The World Wide Web consortium, with industry heavy-weights such as IBM and Microsoft, are providing an alphabet soup of standards: SOAP/XML, WSDL, UDDI and XSDL. »

Stein. Creating a bioinformatics nation. Nature (2002) vol. 417 (6885) pp. 119-20

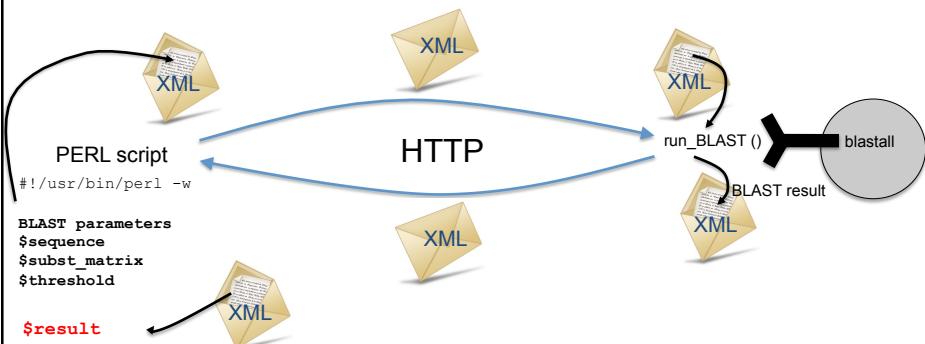
## What are Web Services (WS) ?

- **Definition:**
  - A Web service is a software system designed to support interoperable machine-to-machine interaction over a network

Source: W3C: <http://www.w3.org/TR/ws-gloss/>

## Various types of Web services : SOAP

- SOAP-based Web Services:
  - SOAP: Simple Object Access Protocol
  - Standard of the W3C with specifications: messaging with XML, HTTP for transport



## Various types of Web services : SOAP

**Request envelope**

```

<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:blas="http://tempuri.org/Blast">
  <soapenv:Body>
    <blas:searchParam soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <program xsi:type="xsd:string">blastp</program>
      <database xsi:type="xsd:string">SWISS</database>
      <query xsi:type="xsd:string">MILEGRDGRR</query>
      <param xsi:type="xsd:string"></param>
    </blas:searchParam>
  </soapenv:Body>
</soapenv:Envelope>
  
```

**Perl script**

```

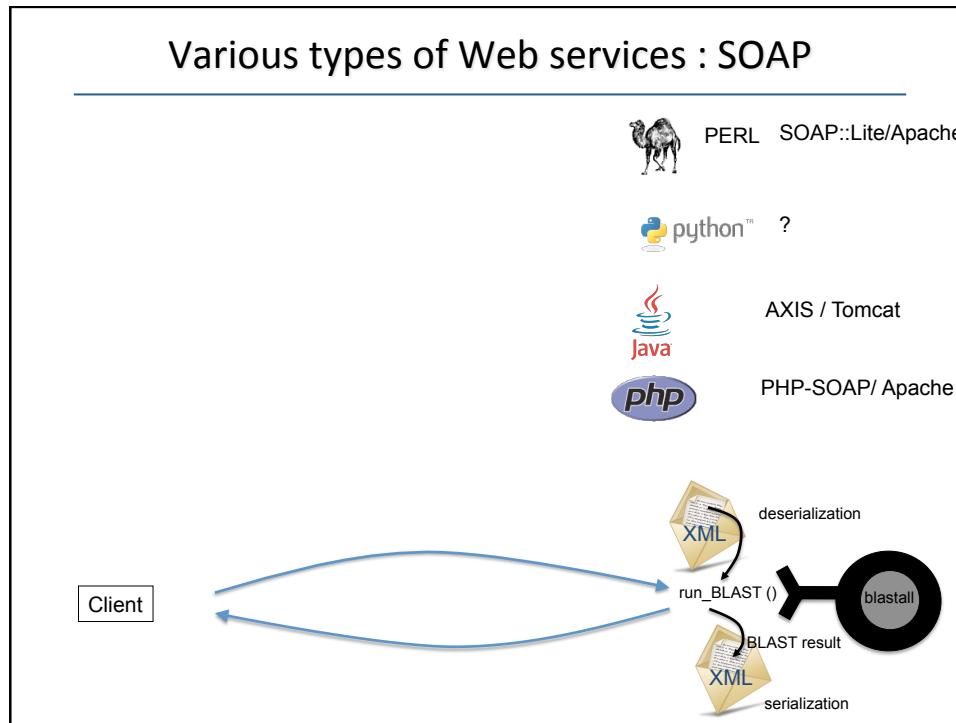
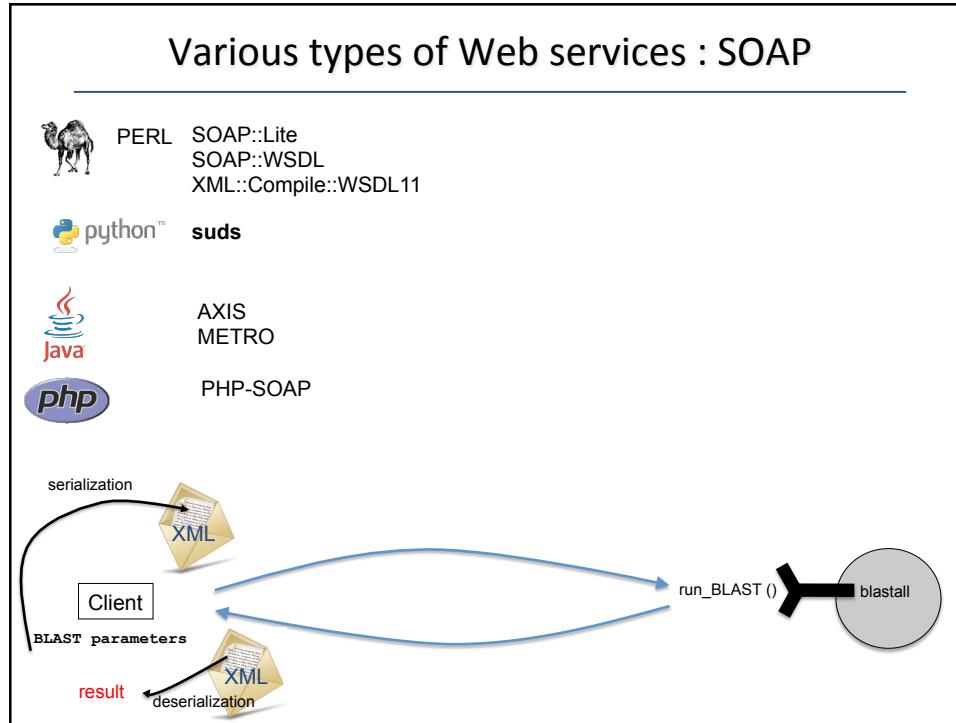
#!/usr/bin/perl -w
  
```

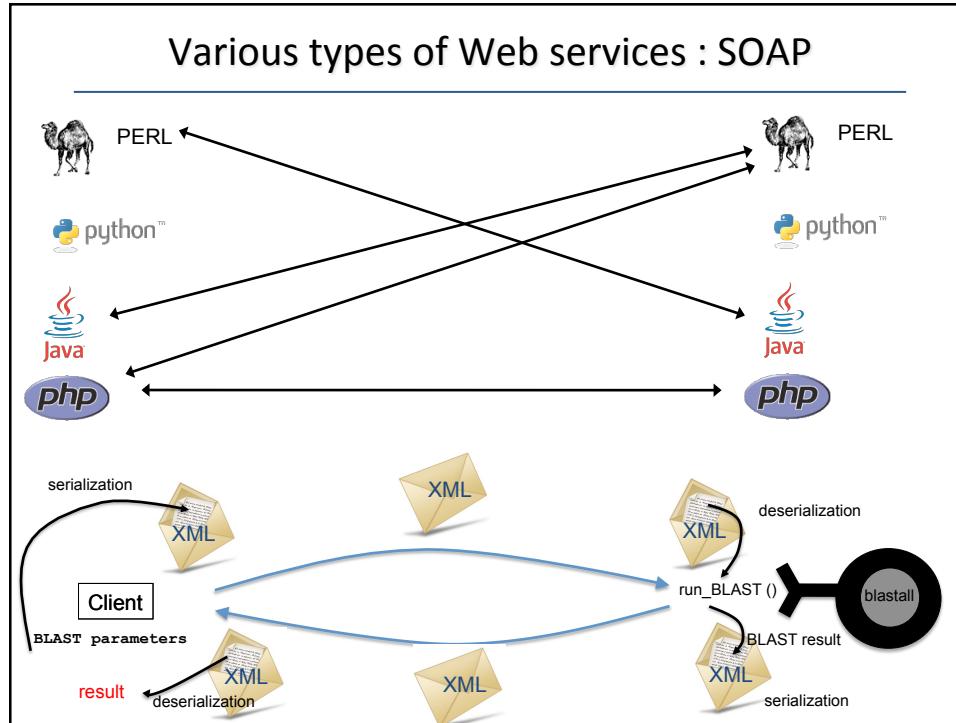
**Response envelope**

```

<soap:Envelope soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <n:searchParamResponse xmlns:n="http://tempuri.org/Blast">
      <Result xsi:type="xsd:string">BLASTP 2.2.18<br/>[Mar-02-2008]</Result>
    </n:searchParamResponse>
  </soap:Body>
</soap:Envelope>
  
```

Reference: Altschul, Stephen F., Thomas L. Madden, Alejandro A. Schaffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997), "Gapped BLAST and PSI-BLAST: a new generation of protein database search





## Various types of Web services : SOAP-WSDL

- WSDL:
  - Web Services Description Language: XML
  - « a machine-readable description of the operations offered by the service »
  - The server « introduce himself » to the clients
    - Names of the available services (=methods)
    - Parameters of each service (name + type)
    - Result of each service (type)

The screenshot shows a web browser displaying a WSDL (Web Services Description Language) document. The document is an XML file with definitions of services, types, and operations. A sidebar on the right contains a navigation menu for 'Web services' with options like 'Introduction', 'Documentation', 'WSDL', 'Clients New !', and 'Taverna workflows'.

```

<?xml version="1.0" encoding ='UTF-8' ?>
<?xmlstylesheet type="text/xsl" href="RSATWS.xsl"?>
<definitions name="RSATWS"
  targetNamespace="urn:RSATWS"
  xmlns:tns="urn:RSATWS"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:html="http://www.w3.org/1999/xhtml"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <types>
    <xsd:schema targetNamespace="urn:RSATWS" xmlns="http://www.w3.org/2001/XMLSchema" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

      <!-- RSA TOOLS REQUESTS -->
      <xsd:complexType name="RetrieveSequenceRequest">
        <xsd:annotation>
          <xsd:documentation>Parameters for the operation retrieve_seq.</xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
          <xsd:element name="output" type="xsd:string" minOccurs="0">
            <xsd:annotation>
              <xsd:documentation>Return type.
Accepted values: 'server' (result is stored on a file on the server), 'client' (result is directly transferred to

```

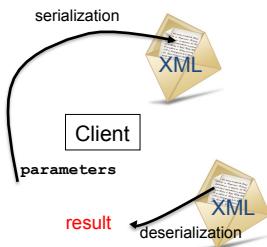
## Various types of Web services : SOAP-WSDL

- WSDL:
  - The URL of the WSDL is necessary to « consume » a SOAP/WSDL Web Service (=write a client)
  - Allows for automatic generation of client-side libraries « client stub »

=> Reduce the amount of code you have to write

**Example:** to write a client for RSAT Web Services in PERL

- SOAP::WSDL installed
- [http://rsat.ulb.ac.be/rsat/web\\_services/RSATWS.wsdl](http://rsat.ulb.ac.be/rsat/web_services/RSATWS.wsdl)
- PERL library « RSATWS » downloadable on RSAT Website, generated from the WSDL



## Various types of Web services : SOAP-WSDL

- Example of code for RSAT PERL Client:

```

#!/usr/bin/perl -w

use SOAP::WSDL;
use lib 'RSATWS';
use MyInterfaces::RSATWebServices::RSATWSPortType;

## new soap object
my $soap=MyInterfaces::RSATWebServices::RSATWSPortType->new();

## parameters
my %args = ('format' => 'text');

## Send the request to the server
my $som = $soap->supported_organisms({'request' => \%args});

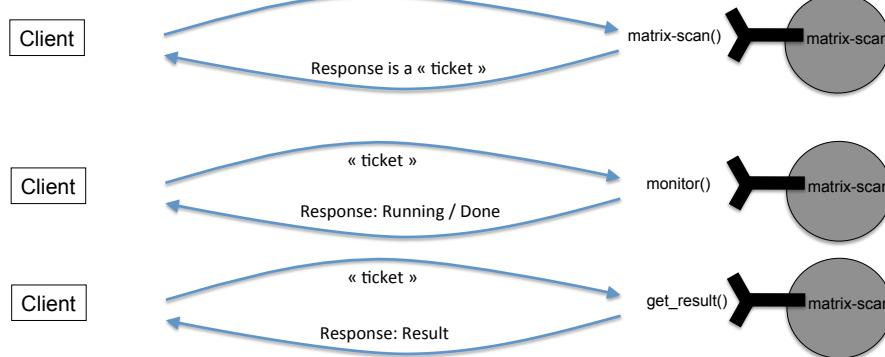
## Get the result
unless ($som) {
    printf "A fault (%s) occurred: %s\n", $som->get_faultcode(), $som->get_faultstring();
} else {
    my $results = $som->get_response();
    my $result = $results -> get_client();
    print "Supported organism(s): %s\n" . $result;
}
  
```

## Various types of Web services : SOAP-WSDL

- Synchrone call



- Asynchrone call (for heavy jobs)



## Various types of Web services : REST

- RESTful Web services:
  - HTTP transport but no messaging system
  - Can be seen as a way to retrieve resources via their URLs
  - Most often used for databases
  - Often not really considered as « Web Services »
- Example:

```

http://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?
db=nucleotide&id=U12345&rettype=fasta
>gi|540023|gb|U12345.1|AMU12345 Apyceros melampus isolate am5 D-loop, partial
sequence; mitochondrial
ACTACCGCTATCAATACTACCTCCACAAATATCAAGAGCCTTCCCAGTATTAAATTGCTAAAATTAA
AATTCAATACGAACCTCACACTCCACGCCAACGGAAATTAAATAACGTATTAAATTCTAGAGTAC
ATACCATGAACCTATCGTTAGTACATGAATTACACACGGTCAGCCGATCAAATGTTTATGTACATAACA
CATTATATATGTCATTTAGTTGTATATAGACATAACATTAAATGTAAAGACATAATAATGTATA
TAGTACATTAATTGATTTGCTCAAGCATATAAGCAAGTACTAGACATTCACTAGGGTACATAGTACAT
TTCAATTGTCATCGTACATAGCGCATGTCAGNCAAATCGTTCTGTCAACATGCATATCCGTCCACTA
GATCAC
  
```

## Web Services: pros and cons

---

- Advantages
  - Independency of languages => interoperability
  - Standard for accessing and describing the services
  - Improved connectivity between the programs
  - Possibility of constructing workflows
- Drawbacks
  - Independency of languages
    - not that straightforward to make a “universal” server
    - Each language has its own “implementation” of the standard
  - Heavy system (SOAP/WSDL), need maintenance by service providers
  - Efficiency => heavy network traffic + serializing/deserializing

## WS everywhere

---

- Amazon
- Google
- Travel agencies....

## WS in Bioinformatics



**Databases** | **Tools**

- about
- ⋮ news
- ⋮ rest
- ⋮ webservices
- ⋮ clients
- ⋮ help
- ⋮ services
- ⋮ soaplab
- ⋮ tutorials
- ⋮ wsdl

<http://www.ebi.ac.uk/Tools/webservices/>



Entrez Programming Utilities  
Updated: August 21, 2008

[http://www.ncbi.nlm.nih.gov/entrez/query/static/eutils\\_help.html](http://www.ncbi.nlm.nih.gov/entrez/query/static/eutils_help.html)



**Web services**

- Introduction
- Documentation
- WSDL
- Clients New !
- Taverna workflows

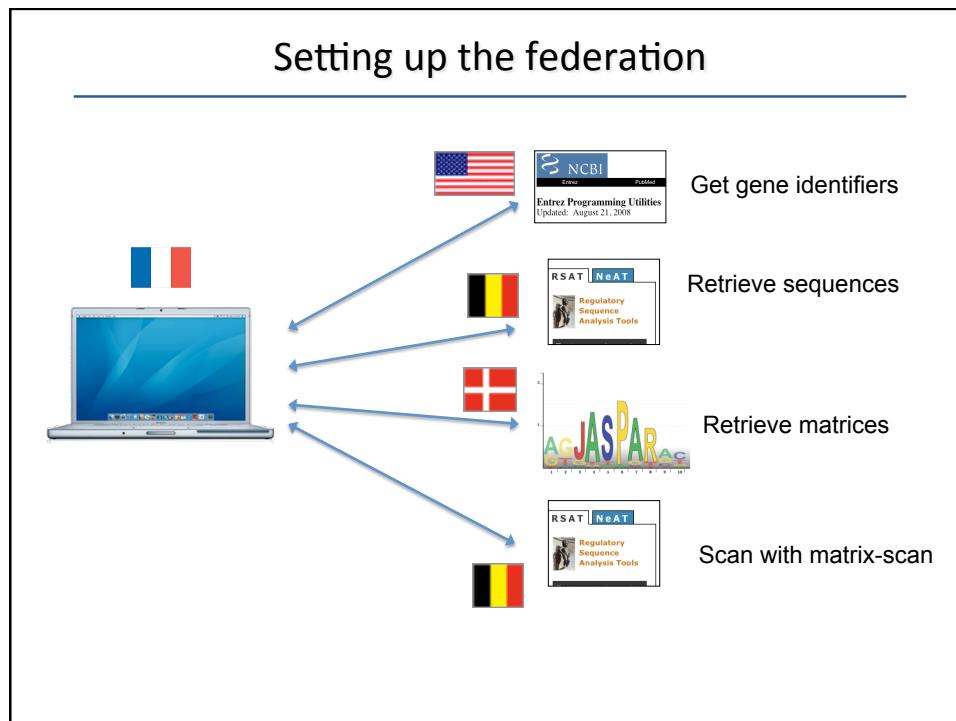
<http://rsat.bigre.ulb.ac.be/rsat/>

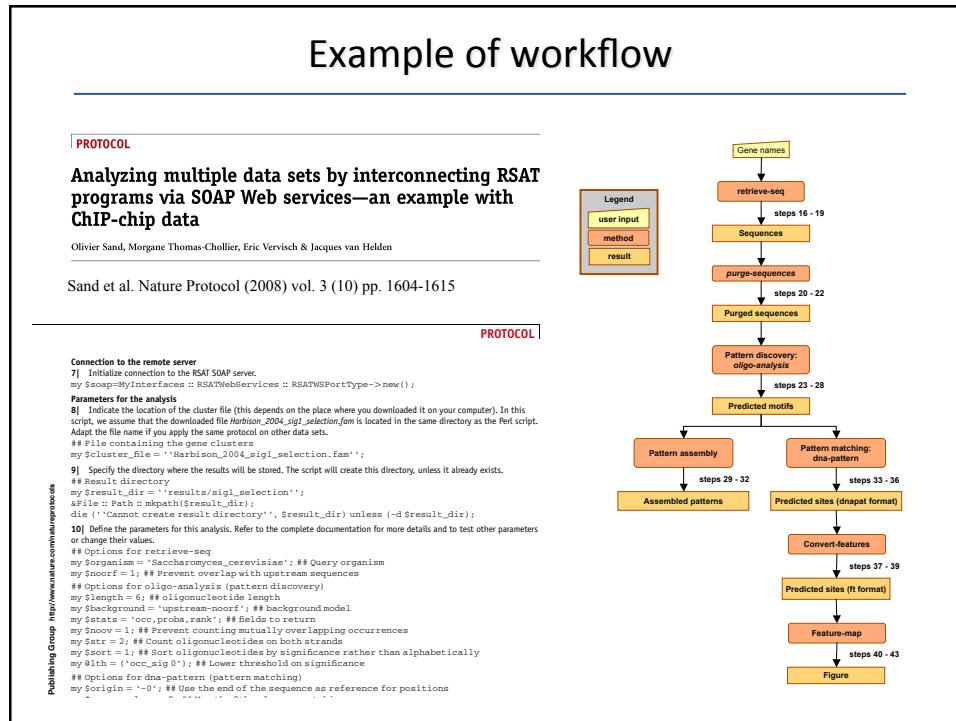


<http://api.bioinfo.no/wsdl/JasperDB.wsdl>



<http://lipm-bioinfo.toulouse.inra.fr/biomoby/>





## Hands on !

---

- Go to the companion website
- Follow all steps of **Session 3**

Morgane Thomas-Chollier