Interoperability of Web Services: granularity and data types

Peter Fischer Hallin, Center for Biological Sequence Analysis
EMBRACE Workshop on Client Side Scripting for Web Services
February 2008
take home from yesterday ...

• are software designed to enable computer-to-computer interaction
• should aim to enhance interoperability
• exchanged objects defined in XSD
• described using WSDL
• commonly exchange data over SOAP/HTTP.
EMBRACE technology

• Services described in WSDL and input/output objects typed using XSD

• Solutions for asynchronous job handling defined using job operations (submit-poll-fetch)

• Encourages to type input/output objects according to the conceptual content

• Services must be documented within Service Descriptions
Outline

• **Design Phase:** Initial considerations for deploying a Web Service

• **Case Story I:** RNAmmer: Consistent annotation of rRNA genes

• **Case Story II:** BLASTatlas; large XML messages and RAW out

• **Exercise:** Preparation for today's exercise: programmatic workflow
Design phase

Making your in-house resource available to the community ...
Which operations will be needed?

- Map the tasks of your resource into I/O of SOAP operation. It may be a workflow which requires multiple operations

- Queuing: If operations take time, a job cannot be handled within a single event of a request-response - queueing (submit-poll-fetch) will be required.
What data is exchanged?

- Make a careful analysis of the conceptual content of your resource.
- Are there existing standards for the specific data you are working with?
- If not, can you find partners with whom you can develop data standards?
- If not, it’s likely you are about to write your own ...
Having the overview, define the data you need to exchange

- SOAP operation I/O is defined using XSD. All the aspects of your data should be defined within the XSD. (arrays, complex data, enumerations, choices, attributes etc)

- XSD is quite a rich format to define XML. There are however rare cases where it’s lacking options. A Rule of thumb: If you take the time, XSD can always fully describe your XML (=no excuses for shortcuts!)
Hooking your data to operations

- This is likely not the difficult part, when all the analysis and XSD writing is done
- Defining the messages from your XSD types, binding them to operations, defining your server endpoint, and composing a WSDL containing everything.
Ensure proper documentation

- EMBRACE defines methods to put documentation at different levels in the WSDL file: All XSD elements can be documented as can the ‘service’ section of your WSDL
documentation

<types>
  <schema>
    <element name="myMethod">
      <annotation>
        <documentation>Datatype elements can be documented in this way. If a wrapper element references other elements that are documented, documentation of the wrapper might not be necessary.</documentation>
      </annotation>
      <complexType>
        <sequence>
          <element name="x" type="int">
            <annotation>
              <documentation>Individual elements can also be documented.</documentation>
            </annotation>
          </element>
          <element name="y" type="float"/>
        </sequence>
      </complexType>
    </element>
    <element name="myMethodResponse">
      <complexType/>
    </element>
  </schema>
  <portType name="PT">
    <operation name="myMethod">
      <documentation>Operations can be documented in this way.</documentation>
      <input message="myMethodRequestMessage"/>
      <output message="empty"/>
    </operation>
  </portType>
</types>
It doesn’t work ...
It doesn’t work ...

• Of course it doesn’t - you haven’t written all the software that connects the server endpoint with your software ...

But this is a different story - and not so relevant to you being the client user...
Case story I: RNAmmer

Tool for predicting rRNA genes in full genome sequences

http://www.psb.ugent.be/rRNA/
Case story I: RNAmmer
Tool for predicting rRNA genes in full genome sequences
Performance: selectivity and sensitivity in the range 0.98-1.00:
The RNAmmer program is available as a traditional HTML-based prediction server at http://www.cbs.dtu.dk/services/RNAmmer as well as through a SOAP-based web service. It is also available for download through the same site.

SUPPLEMENTARY DATA
Supplementary Data is available at NAR online.

ACKNOWLEDGEMENTS
We are grateful for funding from EMBIO at the University of Oslo, the Research Council of Norway and the Danish Center for Scientific Computing. It was also supported by a grant from the European Union through the EMBRACE Network of Excellence, contract number LSHG-CT-2004-512092. We would also like to thank our colleagues for critical reading of the manuscript. Funding to pay the Open Access publication charge was provided by Research Council of Norway.

Conflict of interest statement. None declared.

REFERENCES

Lagesen K, Hallin P, Rodland EA, Staerfeldt HH, Rognes T, Ussery DW
Case-story II: the BLASTatlas WS

A service which allows visualization of homology between a reference genome and any number of genomes, metagenomic samples, or sequence databases.

Example: Seven ocean samples from various depths (surface to 4km): 63,837,557 nucleotides, in 65,674 sequences.

24,978 proteins from 12 fully sequenced Prochlorococcus marinus genomes
green *P. marinus* genomes
blue = ocean samples
Challenges: Massive amounts of input data

- BLASTatlas implemented as a WS, and method was described in a per-review manuscript

- Perl’s SOAP::Lite consumes 10 fold memory during the 20 minutes it take just to prepare the BLASTatlas request. Memory climbs from 200Mb to 2Gb of memory, just doing ... well, we don’t know!

- Bad Web Service design or poor clients?

http://www.cbs.dtu.dk/ws/BLASTatlas
Output is a PostScript document. How does this fit into a SOAP response?

Two options: MIME attachments or encoding the raw file content, and place it in an XML element.
<xsd:complexType name="fetchAtlasResultResponse">
  <xsd:sequence>
    <xsd:element name="output">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="pdf" type="common:EMBRACE document"/>
          <xsd:element name="ps" type="common:EMBRACE document"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="EMBRACE document">
  <xsd:sequence>
    <xsd:element name="comment" type="xsd:string"/>
    <xsd:element name="encoding">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="none"/>
          <xsd:enumeration value="base64"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="MIME type">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="application/pdf"/>  
          <xsd:enumeration value="application/postscript"/>  
          <xsd:enumeration value="application/msword"/>  
          <xsd:enumeration value="application/rtf"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="content" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
XSD allows you to...

- Define (almost) any complex XML structure.
- XSD has build data types for basic content
- `<complexType>` allows you to build your own
- Supports regular expression, enumerations
Data types
Key design considerations

- Common data types
- Granularity
- Typing
Common data types
(if we could all agree ...)

• Common data types could allow seamless connection between different services
• Saves time on documentation and development
• Would however take efforts to administer - and in some cases creating a new type is easier
• Not essential
Granularity, Granularity, and Granularity

• Our choice of technology sets standards for typing any element in input/output - *and these standards should be exploited!*

• To a certain extent, Web Services is all about plumbing - connecting objects (pipes) from different operations to build a workflow and finally to generate a result for you

• This plumbing gets increasingly difficult, the poorer the granularity
Granularity examples:

granularity and object typing

standardization and management
Granularity examples:

A logical threshold for typing: Data that is likely to be a the endpoint of workflows or have limited scientific meaning in the context of Bioinformatics: Jpeg/PNG, PostScript/PDF,
Tropomyosin isoforms

>P09493|TPM1_HUMAN Tropomyosin alpha-1 chain - Homo sapiens (Human).
MDAIKKMKQMLKLKDENAQLDRAEQAEADKKAADRSQKLEDELVSLQKQLKGDDELQKY
SEALKDAQKELAELAEKKAADAVASLNNIRILVEEELDRAQERLALATLQKEEAKEA
DESERGMKVESRAQKDEEMIEIQEQLKEAKHIAEDADRKYEEVARKLVIIESLRAE
ERAELSEGKCAEELLEELKTVNMLKLSEAEQYESQKEKDRYEEEIKVLSKDLKKEABTRA
FAERSVTKLESIDDLEDLYAQKLKIAESEELDDLHNDMSI

>sp_vs|P09493-2|TPM1_HUMAN Isoform 2 of P09493 - Homo sapiens (Human).
MCRLRFRLTASSEHLHERKLPETAEDVSLNNRIQLVEEELDRAQERLALATLQKEEA
EKAADERGMIKQESRAQKDEEMIEIQEQLKEAKHIAEDADRKYEEVARKLVIIESLRAE
ERAELSEGQVRQLEELQRIMDLSESINAEKDYSQKEKDRYEEEIKVLSKDLKKEABTRA
TRAFAERSVTKLESIDDLEDVKVAHAEENLSMHHMLDTQLLELNMA

>sp_vs|P09493-3|TPM1_HUMAN Isoform 3 of P09493 - Homo sapiens (Human).
MDAIKKMKQMLKLKDENAQLDRAEQAEADKKAADRSQKLEDELVSLQKQLKGDDELQKY
SEALKDAQKELAELAEKKAADAVASLNNIRILVEEELDRAQERLALATLQKEEAKEA
DESERGMKVESRAQKDEEMIEIQEQLKEAKHIAEDADRKYEEVARKLVIIESLRAE
ERAELSEGQVRQLEELQRIMDLSESINAEKDYSQKEKDRYEEEIKVLSKDLKKEABTRA
FAERSVTKLESIDDLEDLYAQKLKIAESEELDDLHNDMSI

>sp_vs|P09493-4|TPM1_HUMAN Isoform 4 of P09493 - Homo sapiens (Human).
MDAIKKMKQMLKLKDENAQLDRAEQAEADKKAADRSQKLEDELVSLQKQLKGDDELQKY
SEALKDAQKELAELAEKKAADAVASLNNIRILVEEELDRAQERLALATLQKEEAKEA
DESERGMKVESRAQKDEEMIEIQEQLKEAKHIAEDADRKYEEVARKLVIIESLRAE
ERAELSEGQVRQLEELQRIMDLSESINAEKDYSQKEKDRYEEEIKVLSKDLKKEABTRA
FAERSVTKLESIDDLEDLYAQKLKIAESEELDDLHNDMSI

>sp_vs|P09493-5|TPM1_HUMAN Isoform 5 of P09493 - Homo sapiens (Human).
MAGSSELEAVRKRKLQSAQAADHERRAGTQLRLEHELKLPETAEDVSLNNRIQLVE
EELDRAQERLALATLQKEEAADERGMIKQESRAQKDEEMIEIQEQLKEAKHIAE
DADRKYEEVARKLVIIESLRAEERAELSEGKCAEELLEELKTVNMLKLSEAEQYESQ
KEKDRYEEEIKVLSKDLKKEABTRAFAERSVTKLESIDDLEDQLYQQEQNRRNLNKL
ALNED
<sequences>

&gt; P09493-1 HUMAN Tropomyosin alpha-1 chain - Homo sapiens (Human).
MDAIIKKQMQLKLDKENALDRAEQAEADKKAEDRSKRQLEDELVSLQKLLKGTEDLDKY
SEALKDAQKEKALEAKATDAEADVASLNRINQLVEELDDRAQERLATALKLEAEKAA
DESERNMKVIESRASKDEEKMEIQIEQLKEAKHIAEDADRYEVAVKLVIESDLERAE
ERAELSGKCAELEELKTVTNNLKSLEAQAEKYSQKEDRYEEISIKVLSDKLEAEATRAE
FAERSVTKEKISDILEDQYAQKLKYAIIEELDHALNMTS

&gt; sp_vs|P09493-2 HUMAN Isoform 2 of P09493 - Homo sapiens (Human).
MCRLRIFEXSEHHERKLETAEDVASELNRINQLVEELDDRAQERLATALKLEEA
EKAADRESERGKVIESRAQKDEBEKMBIEIQIEQLEAKHIAEDADRYEVARKLVIESDL
ERAERARSELQVRQEQLREQLIAEDKYSQKEDRYEEISIKVLSDKLEAE
TRAESFASVTKEKISDILEEVAKENLSMHQMEDQTLLELNMM

&gt; sp_vs|P09493-3 HUMAN Isoform 3 of P09493 - Homo sapiens (Human).
MDAIIKKQMQLKLDKENALDRAEQAEADKKAEDRSKRQLEDELVSLQKLLKGTEDLDKY
SEALKDAQKEKALEAKATDAEADVASLNRINQLVEELDDRAQERLATALKLEAEKAA
DESERNMKVIESRASKDEEKMEIQIEQLKEAKHIAEDADRYEVAVKLVIESDLERAE
ERAELSGQVRQLEQLRMDQTLKALMAEDKYSQKEDRYEEISIKVLSDKLEAEATRAE
FAERSVTKEKISDILEDQYAQKLKYAIIEELDHALNMTS

&gt; sp_vs|P09493-4 HUMAN Isoform 4 of P09493 - Homo sapiens (Human).
MDAIIKKQMQLKLDKENALDRAEQAEADKKAEDRSKRQLEDELVSLQKLLKGTEDLDKY
SEALKDAQKEKALEAKATDAEADVASLNRINQLVEELDDRAQERLATALKLEAEKAA
DESERNMKVIESRASKDEEKMEIQIEQLKEAKHIAEDADRYEVAVKLVIESDLERAE
ERAELSGQVRQLEQLRMDQTLKALMAEDKYSQKEDRYEEISIKVLSDKLEAEATRAE
FAERSVTKEKISDILEDQYAQKLKYAIIEELDHALNMTS

&gt; sp_vs|P09493-5 HUMAN Isoform 5 of P09493 - Homo sapiens (Human).
MAGSSSLEAVRKRKSLSEQQAADAERACTLQREDHERKLETAEDVASELNRINQLVE
EELDDRAQERLATALKLEAEKAADESERGKVIESRASKDEBEKMBIEIQIEQLKEAKHIAE
DADRYEVAVKLVIESDLERAEASRELSEGECAELEELKTVNKLKSLAEAQAEKYSQKEDRYEEISIKVLSDKLEAEATRAEFAERSVTKEKISDILEDQYQQLEQNNRRLNKLALNED
</sequence>
Tropomyosin isoforms

<sequences>
  <seq>
    &gt;P09493|TPM1_HUMAN Tropomyosin alpha-1 chain - Homo sapiens (Human).
    MDAIKKKMQMLKLDKENALDRAEQAEADKKAEDRSKQLEDELVSLQKLLKGEDELDKY
    SEALKDAQEKELEAEEKATDAEADVLSRNRIQLVVEELDRAQERLATALQKLLEAEKAA
    DESERGKMVIESRDAQKDEEKMEIQEIQLKEAKHIAEDADRKYEEVARKLVIESDLERAEE
    ERAESEGKCAELEEEELKTVNWLKLSLEAQAEEKYSQKEDRYEEEIKVLSDKLKEABTRAER
    FAERSVTKLEKSIDDELEYAQLKLYKASIAEELDHALNDMSI
  </seq>

  <seq>
    &gt;sp_vs|P09493-2|TPM1_HUMAN Isoform 2 of P09493 - Homo sapiens (Human)
    MCRLRIFLRTASSEHLHERKLRTELAEADVASNLRRQQLVVEELDRAQERLATALQKLEEA
    EKAADESERGKMVIESRDAQKDEEKMEIQEIQLKEAKHIAEDADRKYEEVARKLVIESDL
    ERAEERAESSEGQVRQLEQLRIMDSDELINSNAEDKYSQKEDRYEEEIKVLSDKLKEAE
    TRAEFAERSVTKLEKSIDDELEYVAHAKEENLSMHQMLDQTLLELNNM
  </seq>

  <seq>
    &gt;sp_vs|P09493-3|TPM1_HUMAN Isoform 3 of P09493 - Homo sapiens (Human)
    MDAIKKKMQMLKLDKENALDRAEQAEADKKAEDRSKQLEDELVSLQKLLKGEDELDKY
    SEALKDAQEKELEAEEKATDAEADVLSRNRIQLVVEELDRAQERLATALQKLLEAEKAA
    DESERGKMVIESRDAQKDEEKMEIQEIQLKEAKHIAEDADRKYEEVARKLVIESDLERAEE
    ERAESEGQVRQLEQLRIMDTQITKALMAEDKYSQKEDRYEEEIKVLSDKLKEABTRAER
    FAERSVTKLEKSIDDELEYVAHAKEENLSMHQMLDQTLLELNNM
  </seq>
  <!-- ... -->
</sequences>
Tropomyosin isoforms

<sequences>
  <seg>
    <id>P09493|TPM1_HUMAN Tropomyosin alpha-1 chain - Homo sapiens (Human).</id>
    <sequence>MDAIKKKMQLKLDKENALDRAEQAEADKKAAEDRRSKQLEDELVSLQKLLKGTEDELDKYSEALKDAQEKLELAEEKKATDAEADV</sequence>
  </seg>
  <id>sp_vs|P09493-2|TPM1_HUMAN Isoform 2 of P09493 - Homo sapiens (Human)</id>
  <sequence>MCRRLRFLRTASSEHLHERKLRETAADVASNRRILQLVVEELDRAQRLATALQKLEEAEEKAAADESERGKVIESRSAQKDEE</sequence>
  <id>sp_vs|P09493-3|TPM1_HUMAN Isoform 3 of P09493 - Homo sapiens (Human)</id>
  <sequence>MDAIKKKMQLKLDKENALDRAEQAEADKKAAEDRRSKQLEDELVSLQKLLKGTEDELDKYSEALKDAQEKLELAEEKKATDAEADV</sequence>
  <id>sp_vs|P09493-4|TPM1_HUMAN Isoform 4 of P09493 - Homo sapiens (Human)</id>
  <sequence>MDAIKKKMQLKLDKENALDRAEQAEADKKAAEDRRSKQLEDELVSLQKLLKGTEDELDKYSEALKDAQEKLELAEEKKATDAEADV</sequence>
  <id>sp_vs|P09493-5|TPM1_HUMAN Isoform 5 of P09493 - Homo sapiens (Human)</id>
  <sequence>MAGSSSLEAVRRKIRSLQEQADAASEERAGTLQRELDHERKLRETAADVASNRRILQLVVEELDRAQRLATALQKLEEA</sequence>
</sequences>
A ridiculous example: Substance P (a neuro transmitter)
A ridiculous example:
Substance P (a neuro transmitter)

<protein>
  <id>Substance P</id>
  <UniProt>NM_003182</UniProt>
  <sequence>
    <unit>
      <formula tla="Arg">
        <atom name="C" count="6"/>
        <atom name="H" count="14"/>
        <atom name="N" count="4"/>
        <atom name="O" count="2"/>
      </formula>
    </unit>
    <unit>
      <formula tla="Pro">
        <atom name="C" count="5"/>
        <atom name="H" count="9"/>
        <atom name="N" count="1"/>
        <atom name="O" count="1"/>
      </formula>
    </unit>
    <unit>
      <formula tla="Lys">
        <atom name="C" count="6"/>
        <atom name="H" count="14"/>
        <atom name="N" count="2"/>
        <atom name="O" count="2"/>
      </formula>
    </unit>
    <unit>
      <formula tla="Pro">
        <atom name="C" count="5"/>
        <atom name="H" count="9"/>
        <atom name="N" count="1"/>
        <atom name="O" count="1"/>
      </formula>
    </unit>
  </sequence>
</protein>
A ridiculous example:

Substance P (a neuro transmitter)
A ridiculous example:

Substance P (a neuro transmitter)
A ridiculous example:
Substance P (a neuro transmitter)
Typing
Typing

array

id: string

sequence: string - restrictions?

/^[ACDEFGHIKLMNPQRSTVWY]+$/
Today's Exercise

(warning: a little bit of biology ahead)
Sigma-70 transcription factor is a protein which binds -10 and -35 nt upstream of the transcription start site during the prokaryotic gene transcription.

Sigma factor facilitates transcription by binding to the RNA polymerase

Sigma factors serve a regulatory function

Aim for the exercise is to predict the promoter binding sites of sigma factors, more specifically looking at the P1 and P2 promoters for ribosomal RNA genes.
**E. coli rRNA operons**

- *E. coli* has seven rRNA operons, *rrnB* being the most intensively studied.
- All rRNA genes of 16S, 23S, and 5S plus at least one tRNA gene are encoded on the same transcript.
- Regulated by two promoters P1 and P2, with a near-consensus $\sigma^{70}$ core promoter regions.
- P1 is stronger during fast growth rates, whereas transcription from P2 is predominant at slow rates or during prolonged stationary phase.

![Diagram of E. coli rRNA operons](image-url)
The *E. coli* rrnB operons

- btuB
- murI
- 16S
- tRNA\textsuperscript{Glu}
- 23S
- 5S
- murB

**Position relative to 16S gene start**

- P1: Raw combined scores
- P2: Adjusted combined scores

**Results**

- P1: Raw combined scores, P2: Adjusted combined scores

- Estrem et al. 1998
- Huerta et al. 2003
- Hengen et al. 1997

**Figures**

- (a) 400
- (b) 35, UP (E. coli) (N=63)
- (c) 5
- (d) 35, UP (E. coli) (N=63)
tRNA and rRNA prediction

• Prediction of tRNA genes using tRNAscan (Lowe et al. 1997) and RNAmmer (Lagesen et al. 2007)


http://lowelab.ucsc.edu/tRNAscan-SE/


Y-proteobacteria

Figure 2.2: Neighbor-Joining tree of first 1k bases of all 16S rRNA genes of *Yersinia, Salmonella, Shigella, and E. coli*
General $\sigma^{70}$ core, FIS site, and UP element (E. coli)

Figure 2.4: Logo plots showing the initial weight matrices used for searching E. coli genomes. $-10$ hexamer (a), $-35$ hexamer (b), UP element (c), and FIS binding motif (d).

By using information theory one can compose a measure of how well a given aligned query sequence conforms to a given weight matrix, measured in bits of information. By producing a matrix of $R_{b,p}$ values, (see equation 3.3) the $R_i$ value is obtained by aligning the query sequence to the matrix columns and summarize the $R_{b,p}$ values.

$$R_{b,p} = \log_2(4) + \log_2 \frac{n_{b,p}}{N} \quad R_{tot} = \sum_{p=1}^{L} R_{B,p} \quad (3.3)$$

$-\text{where } b \in ATGC \text{ iterates through the four bases, } p \text{ denotes the position in the alignment, } L \text{ is the length of the alignment (or width of the matrix), and } n_{b,p} \text{ is the number of bases } b \text{ at position } p, \text{ and } B \text{ denotes the base at position } p \text{ in the query}$
Spacing and scoring of multiple motifs

Recently, a method was proposed by Shultzaberger and co-workers introducing an information theory based measure, to quantify the helical phasing of adjacent binding motifs (Shultzaberger et al., 2007). The authors defines an accessibility, \( n(d) \), equation 3.4, and gap surprisal, \( GS(d) \).

\[
n(d) = 1 + \cos \left( \frac{2\pi}{w}(d - c) \right)
\]  

--- accessibility ---

–where \( c \) is the center distance between two binding sites (e.g. optimally spaced), \( d \) is the query distance, \( w = 10.6 \) is the distance of a one helix turn of B-form DNA. Finally, this gives the \( GS(d) \) as follows:

\[
GS(d) = \log_2 \frac{n(d)}{N}
\]  

--- distance score (bits) ---

–where \( N \) is the sum of all \( n(d) \), defined in equation 3.6. The sign of the \( GS(d) \) was changed from the original equation described (Shultzaberger et al., 2007) to allow for combining all scores by addition.

\[
N = \sum_{d=\text{min}}^{\text{max}} n(d)
\]  

--- sum of all \( n(d) \) ---

–where \( \text{min} \) and \( \text{max} \) are the boundaries of a given window examined. The program iscan was written based on the framework of the gap surprisal, helical accessibility, and individual information based weight matrices just described. The program supports any number of PWMs, separated by user-defined spacers applying the \( GS(d) \) measure.

\[
R_i(\text{tot}) = R_i(m_1) + GS(d, m_1) + R_i(m_2) + \ldots + GS(d, m_{n-1}) + R_i(m_n)
\]  

--- Total bit score of all boxes ---
*Initial combined $\sigma^{70}$ core, FIS I site, and UP element models (E. coli and Shigella)*

![Graphs showing initial combined $\sigma^{70}$ core, FIS I site, and UP element models](image)

Figure 2.5: Profiles showing the *iscan* scores of the initial weight matrices (see section 2.1.2) applied to *E. coli* and *Shigella*: Unadjusted P1 scores (a), Adjusted P1 scores (b), Unadjusted P2 scores (c), and Adjusted P2 scores (d)
$1^{st}$ iteration: model scores (E.coli+Shigella)

Figure 2.7: Raw and adjusted *iscan* profiles of *E. coli* and *Shigella* using refined P1 and P2 matrices for *E. coli*: Unadjusted P1 scores (a), adjusted P1 scores (b), unadjusted P2 scores (c), and adjusted P2 scores (d)
Strong similarity of P1 and P2 sites
exercise 2

- Locate rRNA genes
- Extract promoter regions
- Predict the presence of P1 and P2
- Extract the -10 and -35 regions
- Build a workflow
Acknowledgements

David W. Ussery
Craig Benham
Karin Lagesen
Jan Christian Bryne
Francisco Roque
Kristoffer Rapacki
Hans Henrik Stærfeldt