Protein-Protein Interactions

Introduction to Data Processing

May 2019

Lukasz Salwinski
lukasz@mbi.ucla.edu
Boyer Hall 205

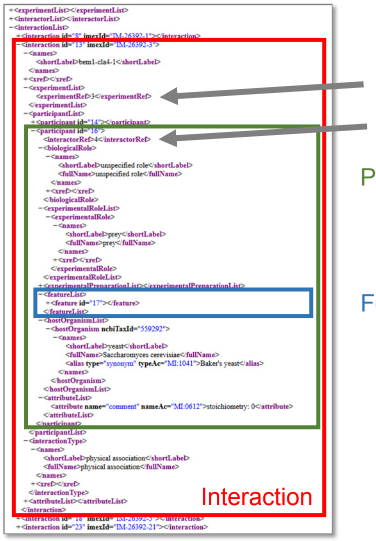
Interaction Data Processing

Topics

- MIF file structure
  - ‘compact vs ‘expanded’ variants
- Binary expansion of multi-protein interactions
  - ‘spoke’ vs ‘matrix’ expansion
- XML parsing
  - Lxml library
  - Xpath
- Extracting data from MIF files
  - Access relevant/useful information
  - Prepare files ready to import into Cytoscape

Interaction Data Processing

PSI-MI XML (MIF) file anatomy



```
+<experimentList></experimentList>
+<interactorList></interactorList>
−<interactionList>
  +<interaction id="8" imexId="IM-26392-1"></interaction>
  −<interaction id="13" imexId="IM-26392-3">
    −<names>
       <shortLabel>bem1-cla4-1</shortLabel>
     </names>
    +<xref></xref>
    −<experimentList>
       <experimentRef>3</experimentRef>
     </experimentList>
    −<participantList>
      +<participant id="14"></participant>
      −<participant id="16">
         <interactorRef>4</interactorRef>
        −<biologicalRole>
          −<names>
             <shortLabel>unspecified role</shortLabel>
             <fullName>unspecified role</fullName>
           </names>
          +<xref></xref>
         </biologicalRole>
        −<experimentalRoleList>
          −<experimentalRole>
            −<names>
               <shortLabel>prey</shortLabel>
               <fullName>prey</fullName>
             </names>
            +<xref></xref>
           </experimentalRole>
         </experimentalRoleList>
        +<experimentalPreparationList></experimentalPreparationList>
        −<featureList>
          +<feature id="17"></feature>
         </featureList>
        −<hostOrganismList>
          −<hostOrganism ncbiTaxId="559292">
            −<names>
               <shortLabel>yeast</shortLabel>
               <fullName>Saccharomyces cerevisiae</fullName>
               <alias type="synonym" typeAc="MI:1041">Baker's yeast</alias>
             </names>
           </hostOrganism>
         </hostOrganismList>
        −<attributeList>
           <attribute name="comment" nameAc="MI:0612">stoichiometry: 0</attribute>
         </attributeList>
       </participant>
     </participantList>
    −<interactionType>
      −<names>
         <shortLabel>physical association</shortLabel>
         <fullName>physical association</fullName>
       </names>
      +<xref></xref>
     </interactionType>
    +<attributeList></attributeList>
   </interaction>
  +<interaction id="18" imexId="IM-26392-5"></interaction>
  +<interaction id="23" imexId="IM-26392-21"></interaction>
```

Compact MIFs use references to Interactors and/or Experiments

Description of the experimental method (explicit or a reference)

Description of the interactor (explicit or a reference)

Participant (e.g. protein)

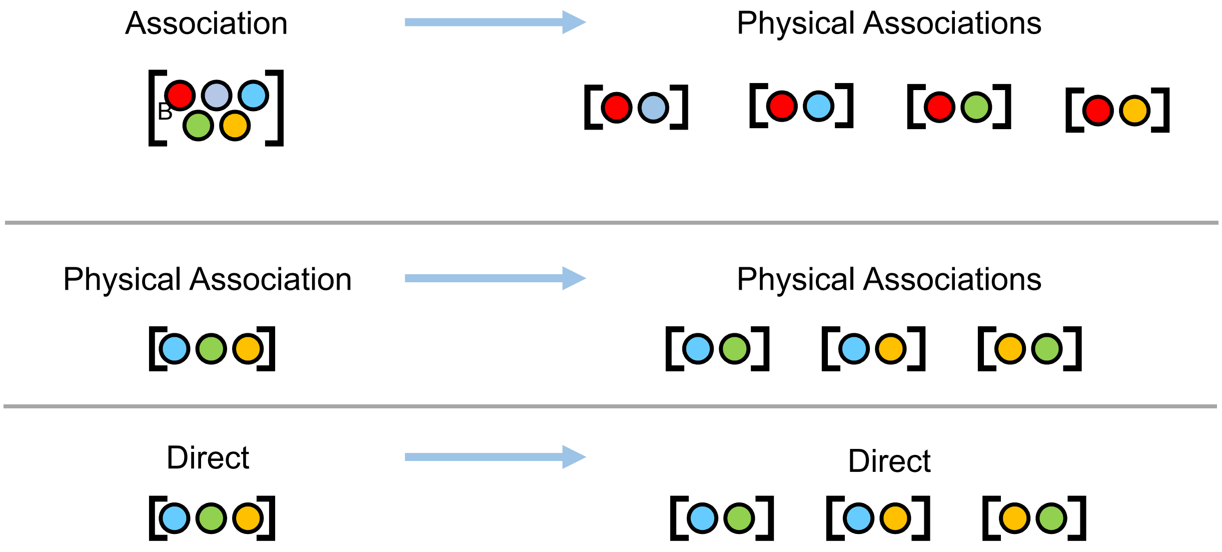
- Describes the state of the molecule as used in the experiment
- Refers to ‘interactor’ – the reference description of the molecule (e.g. UniprotKB)

Features (e.g. mutations)

- Modifications of the molecule relative to the reference state

Interaction Data Processing

Binary Expansion



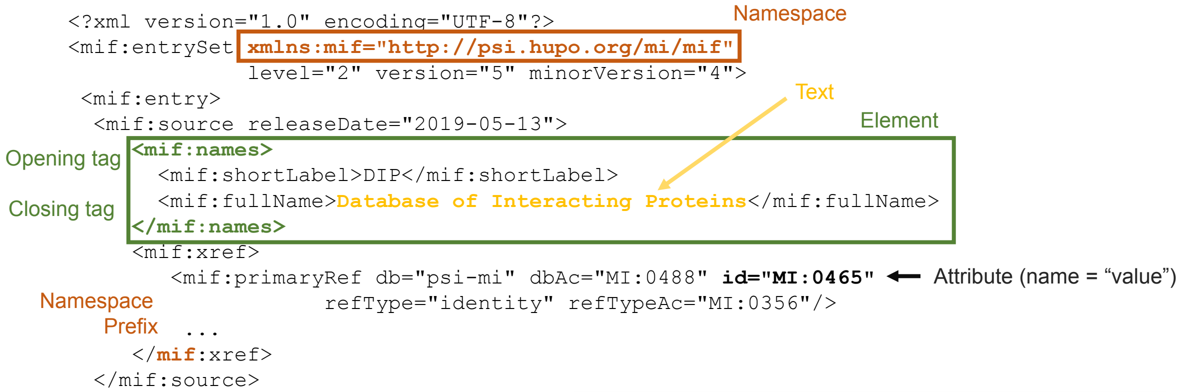
Association → Physical Associations

Physical Association → Physical Associations

Direct → Direct

Interaction Record Formats

XML file anatomy



```
<?xml version="1.0" encoding="UTF-8"?>
<mif:entrySet xmlns:mif="http://psi.hupo.org/mi/mif"
              level="2" version="5" minorVersion="4">
 <mif:entry>
  <mif:source releaseDate="2019-05-13">
    <mif:names>
      <mif:shortLabel>DIP</mif:shortLabel>
      <mif:fullName>Database of Interacting Proteins</mif:fullName>
    </mif:names>
    <mif:xref>
       <mif:primaryRef db="psi-mi" dbAc="MI:0488" id="MI:0465"
                   refType="identity" refTypeAc="MI:0356"/>
       ...
    </mif:xref>
  </mif:source>
   ...
 </mif:entry>
</mif:entrySet>
```

<element attribute="value"/>
is equivalent to
<element attribute="value"></element>

Interaction Data Processing

Lxml library (https://lxml.de/)

```
from lxml import etree
from io import StringIO
```

Only if needed...

```
xml = '<protein acc="P60010"><seq>MKYDDEW...</seq></protein>'
```

```
strDom = etree.fromstring( xml )
              or
strDom = etree.fromstring( StringIO(xml) )
```

Parse String

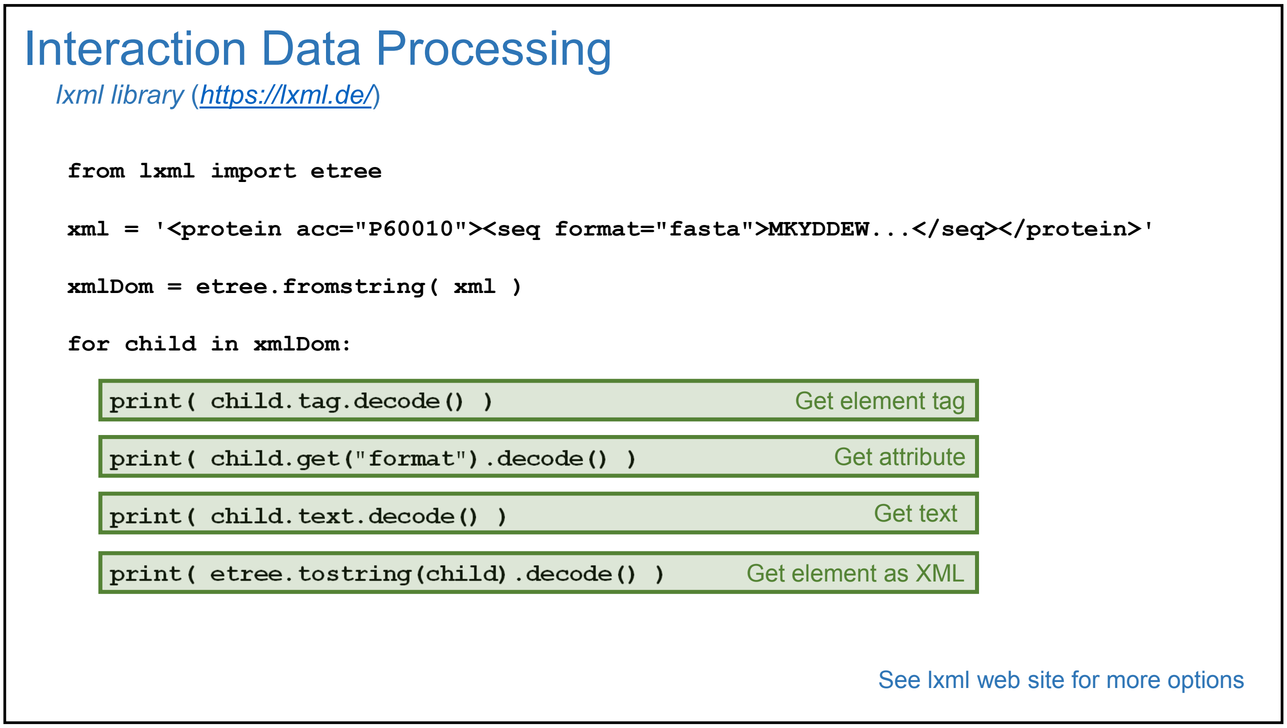
```
fileDom = etree.parse("doc/test.xml")
              or
fileDom = etree.parse( open("doc/test.xml") )
```

Parse File/URL
.gz files OK

```
print( etree.tostring( strDom ).decode())
```

... and back to String

See lxml web site for more options

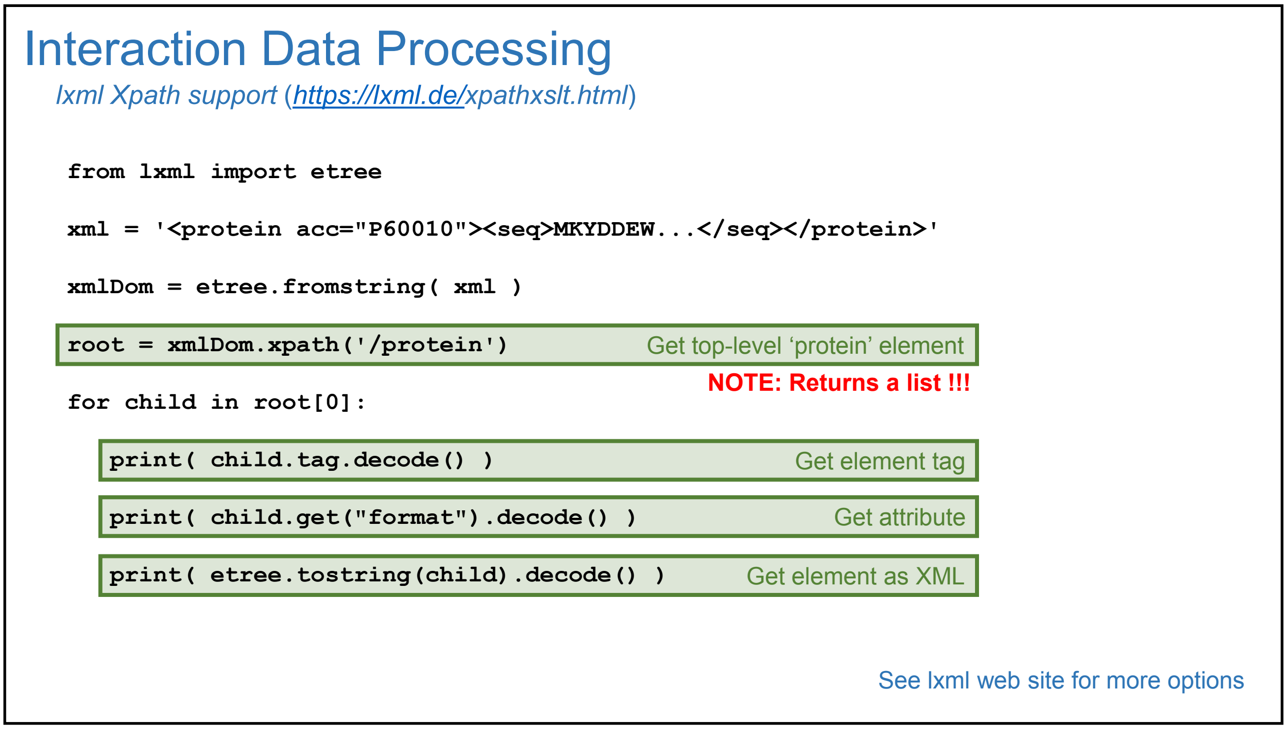


Interaction Data Processing

lxml library (https://lxml.de/)

```
from lxml import etree

xml = '<protein acc="P60010"><seq format="fasta">MKYDDEW...</seq></protein>'

xmlDom = etree.fromstring( xml )

for child in xmlDom:

   print( child.tag.decode() )                           Get element tag

   print( child.get("format").decode() )                 Get attribute

   print( child.text.decode() )                          Get text

   print( etree.tostring(child).decode() )               Get element as XML
```

See lxml web site for more options



Interaction Data Processing

lxml Xpath support (https://lxml.de/xpathxslt.html)

```
from lxml import etree

xml = '<protein acc="P60010"><seq>MKYDDEW...</seq></protein>'

xmlDom = etree.fromstring( xml )

root = xmlDom.xpath('/protein')              Get top-level ‘protein’ element
                                                   NOTE: Returns a list !!!
for child in root[0]:

   print( child.tag.decode() )                           Get element tag

   print( child.get("format").decode() )                 Get attribute

   print( etree.tostring(child).decode() )               Get element as XML
```

See lxml web site for more options

Interaction Data Processing

lxml Xpath support (https://lxml.de/xpathxslt.html)

```
from lxml import etree

xml = '<protein acc="P60010"><seq>MKYDDEW...</seq></protein>'

xmlDom = etree.fromstring( xml )
```

```
t1 = xmlDom.xpath('/protein/seq/text()')
```

Get the text of 'seq' elements that are children of the top-level 'protein' element

```
t2 = xmlDom.xpath('//seq/text()')
```

Get the text of ANY 'seq' element

```
e3 = xmlDom.xpath('//protein[@acc="P60010"]/seq')
```

Get 'seq' elements that are children of 'protein' element with 'acc' attribute equals to 'P60010'

```
e4 = e3[0].xpath(‘./text()’)
```

Get the text of the current element

See *https://en.wikipedia.org/wiki/XPath* and *https://www.w3.org/TR/xpath-10* for more details

Interaction Data Processing

lxml Xpath namespace support (https://lxml.de/xpathxslt.html)

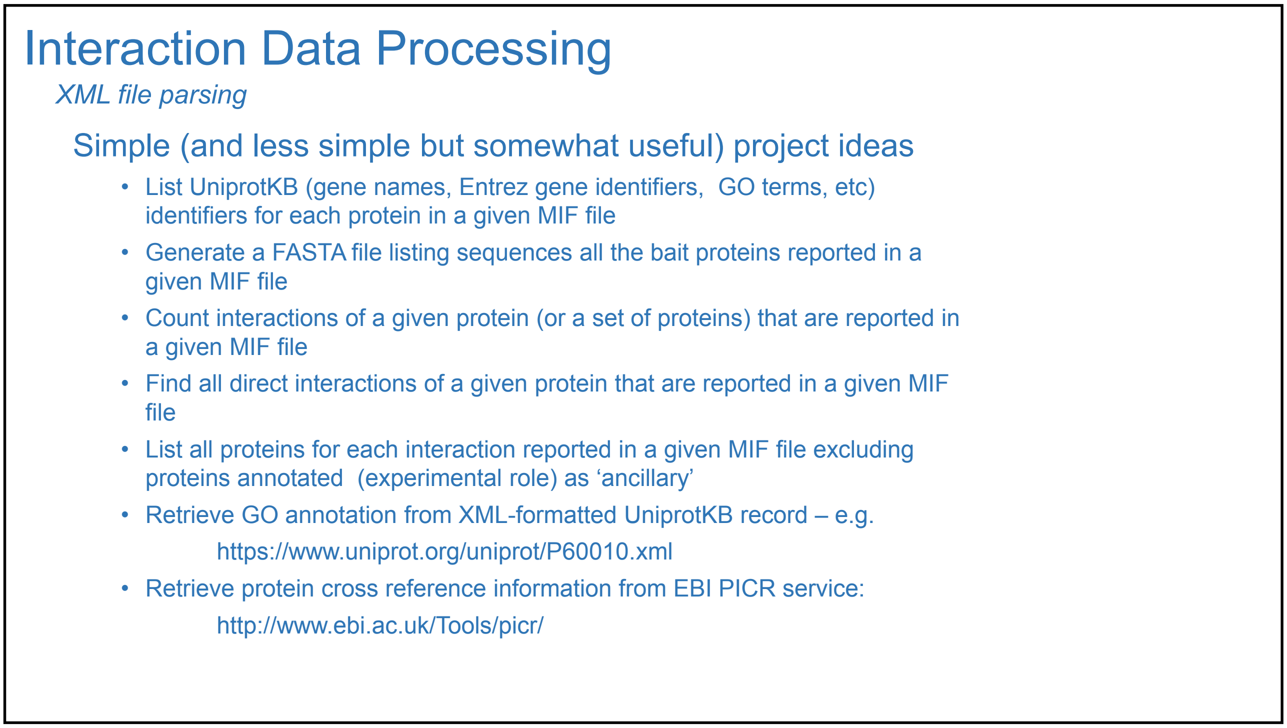
```
from lxml import etree

xml = '''<mif:protein xmlns:mif="http://psi.hupo.org/mi/mif" acc="P60010">
          <mif:seq>MKYDDEW...</mif:seq>
       </mif:protein>'''

xmlDom = etree.fromstring( xml )

e = xmlDom.xpath('/m:protein/m:seq',
                 namespaces={'m': 'http://psi.hupo.org/mi/mif'})
```

```
print( e[0].tag.decode())
```

Get qualified (i.e. with namespace) tag

```
qname = etree.QName(e[0])

print( qname.localname.decode() )
print( qname.namespace.decode() )
```

Split qualified tag into namespace and local name



Interaction Data Processing

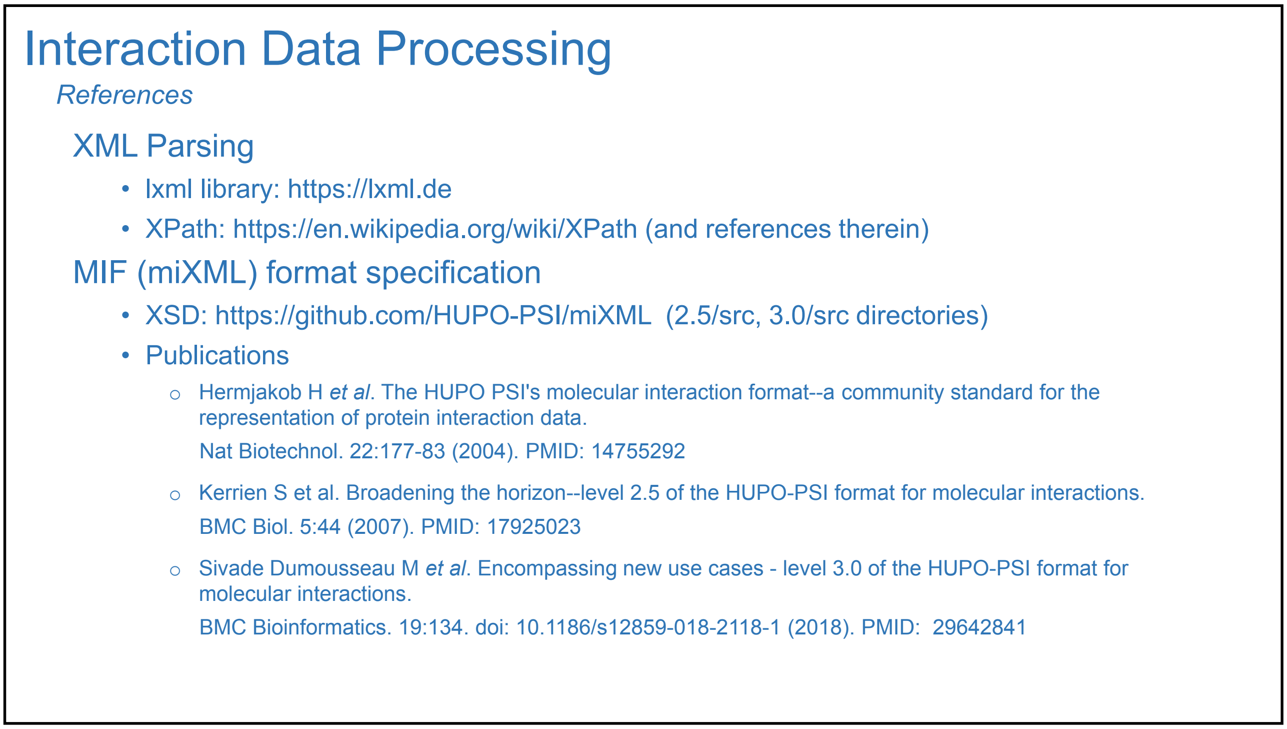
XML file parsing

Simple (and less simple but somewhat useful) project ideas

- List UniprotKB (gene names, Entrez gene identifiers, GO terms, etc) identifiers for each protein in a given MIF file
- Generate a FASTA file listing sequences all the bait proteins reported in a given MIF file
- Count interactions of a given protein (or a set of proteins) that are reported in a given MIF file
- Find all direct interactions of a given protein that are reported in a given MIF file
- List all proteins for each interaction reported in a given MIF file excluding proteins annotated (experimental role) as ‘ancillary’
- Retrieve GO annotation from XML-formatted UniprotKB record – e.g.

  https://www.uniprot.org/uniprot/P60010.xml
- Retrieve protein cross reference information from EBI PICR service:

  http://www.ebi.ac.uk/Tools/picr/



Interaction Data Processing

References

XML Parsing

- lxml library: https://lxml.de
- XPath: https://en.wikipedia.org/wiki/XPath (and references therein)

MIF (miXML) format specification

- XSD: https://github.com/HUPO-PSI/miXML (2.5/src, 3.0/src directories)
- Publications
  - Hermjakob H *et al*. The HUPO PSI's molecular interaction format--a community standard for the representation of protein interaction data.

    Nat Biotechnol. 22:177-83 (2004). PMID: 14755292
  - Kerrien S et al. Broadening the horizon--level 2.5 of the HUPO-PSI format for molecular interactions.

    BMC Biol. 5:44 (2007). PMID: 17925023
  - Sivade Dumousseau M *et al*. Encompassing new use cases - level 3.0 of the HUPO-PSI format for molecular interactions.

    BMC Bioinformatics. 19:134. doi: 10.1186/s12859-018-2118-1 (2018). PMID: 29642841