UGENE Workflow Designer.
Flexible control and extension of pipelines with scripts

Yuriy Vaskin
INTEGRATION is the NEW SEXY
Software development in Bioinformatics

Yeah! I’ve come up with a new tool

Ooh... It’s already there

About 68,700 results (0.27 seconds)
Hey! I need all the tools.
• Multiple Alignment: MUSCLE, Kalign, Clustal, Mafft, T-Coffee
• Ultra fast repeat finder

• HMMER, ORF Finder, Restriction sites
• Remote databases queries (BLAST, CDD, DAS, etc.)
• Phylogenetic trees
• TFBSs
• > 20 data formats
UGENE Workflow Designer Features

- On **local** machine
- Share files with schemes
- Imports data to internal DB
- Or passes file URLs
- Extension with external instruments
Wizards to setup the parameters

SAMtools mpileup parameters

- Count anomalous read pairs: False
- Disable BAQ computation: False
- Mapping quality downgrading coefficient: 0
- Max number of reads per input BAM: 250
- Extended BAQ computation: False
- BED or position list file: 
- Pileup region: 
- Minimum mapping quality: 0
- Minimum base quality: 13
- Additional parameters: 

[Show additional parameters]
Dashboard for results and statistics

Output Files

<table>
<thead>
<tr>
<th>File</th>
<th>Producer</th>
</tr>
</thead>
<tbody>
<tr>
<td>res_seq.g1</td>
<td>Write Sequence</td>
</tr>
</tbody>
</table>

Workflow Task

Time: 00:00:01

The workflow task has been finished successfully!

Common Statistics

<table>
<thead>
<tr>
<th>Element</th>
<th>Elapsed time</th>
<th>Output messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter</td>
<td>00:00:00</td>
<td>1</td>
</tr>
<tr>
<td>Sequence Marker</td>
<td>00:00:00</td>
<td>1</td>
</tr>
<tr>
<td>Read Sequence</td>
<td>00:00:00</td>
<td>1</td>
</tr>
<tr>
<td>Find Pattern</td>
<td>00:00:00</td>
<td>1</td>
</tr>
<tr>
<td>Write Sequence</td>
<td>00:00:00</td>
<td>0</td>
</tr>
</tbody>
</table>
UGENE Workflow Designer API

Implemented in C

Extended for Python

Let users setup and run workflows
import u2py
u2py.initContext('./')

inputFile = 'data/samples/FASTA/chrUn_gln000231.fa'
outputFile = 'output_ann.gb'

u2py.Scheme.launchSas('repeats-search', inputFile, outputFile)
u2py.releaseContext()
from Bio.Align import MultipleSeqAlignment
from Bio import AlignIO
import u2py
u2py.initContext( './' )

#read an alignment
inputAlignments = []
inputFile = open( '../..//data/samples/CLUSTALW/COI_copy1.sto', 'rU' )
inputAlignments.append( AlignIO.read( inputFile, 'stockholm' ) )
inputFile.close( )

#create a scheme
scheme = u2py.Scheme( 'muscle', inputAlignments )
scheme.setElementAttribute( 'Write Alignment', 'document-format', 'stockholm' )

#run the scheme
outputAlignments = scheme.launch( )

scheme.cleanUp( )
u2py.releaseContext( )