FOIS 2012: 7th International Conference on Formal Ontology in Information Systems

Suggestions for Galaxy Workflow Design Using Semantically Annotated Services

Alok Dhamanaskar, Michael E. Cotterell, Jessica C. Kissinger, and John Miller.
University of Georgia

Jie Zheng and Christian J. Stoeckert, Jr.
University of Pennsylvania

Presented by : Jie Zheng

Outline

1. Web Service Composition Issues
2. Semantic Annotation of Web Services
3. Semi-Automatic Workflow Composition:
   Service Suggestion Engine (SSE)
   - Input-Output Matching Algorithms
   - Objective Specification Compliance
   - Calculation of Concept Similarity
5. SSE Evaluation
   - Interfacing SSE with the Galaxy workflow editor
6. Summary and Future Work

Web Service Workflow Composition Issues

- Web services are developed by different contributors
- Not developed to work with one another

Web services are described using either a WSDL (Web service Description Language) document or a WADL (Web Application Development Language) document.

  - No standard naming conventions, *e.g., operation, input*
  - Text descriptions are inherently ambiguous

Hard to identify whether inputs/outputs are compatible

Semantic Web Service frameworks

- OWL-S
  - Upper level ontology for Web services
  - Top-down approach
  - Difficult to model the large number of existing Web services

- **SAWSDL**
  - Bottom-up approach
  - Provides extension attributes for adding semantics to Web services: **sawsdl:modelReference**
  - Easy to semantically model existing Web services
  - No specific semantic model or ontology language required to use
  - W3C recommended web services semantic annotation mechanism

Bioinformatics Web Service Ontology (OBI-WS)

- OBI WS v 1.0 released
  - http://purl.obolibrary.org/obo/obi/webService.owl
  - Bioportal: http://bioportal.bioontology.org/ontologies/3119
- Supports annotation of ~100 operations from 19 different web services including sequence analysis and utility web services
  - Sequence Similarity Web Services
    - WU-BLAST, NCBI-BLAST
  - Multiple Sequence Alignment Web services
    - Clustal W, T-coffee
  - Protein Functional Analysis Web services
    - SignalP
  - Phylogenetic Analysis Web services
    - Phylip

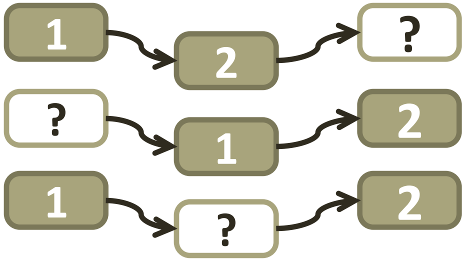
Service Suggestion Engine

- The Service Suggestion Engine (**SSE**) is a semi-automatic workflow composition system.
- **What it does:**
  - Facilitates the construction and extension of workflows by providing **suggestions** to the user for the next step.
  - It is capable of doing

    Forward Suggestions,

    Backward Suggestions and

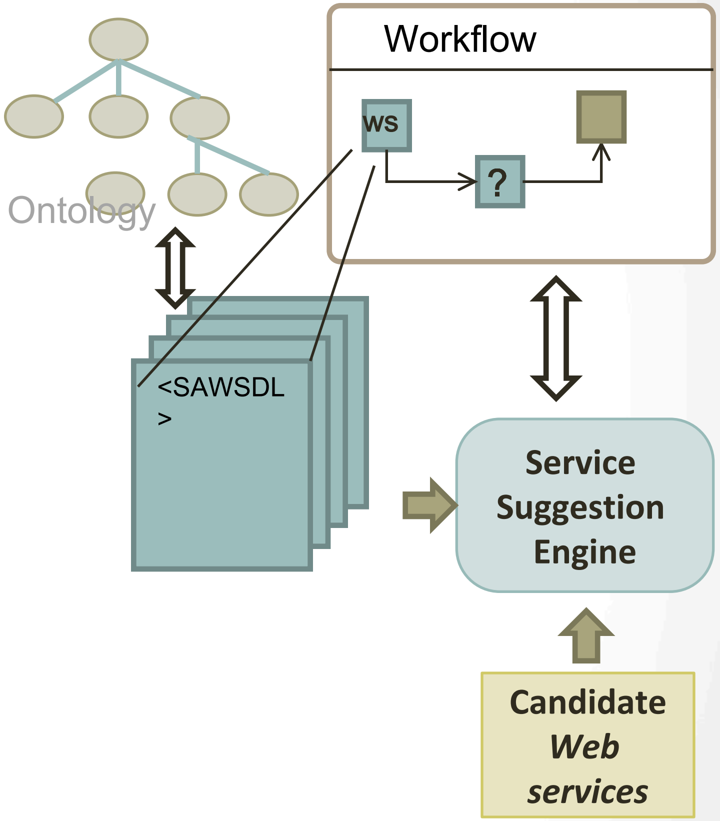
    Bi-directional Suggestions.



  - SSE returns a ranked list of web service operations (from a candidate list of available operations) that could be used for the *next, previous* or *intermediate* step.
  - SSE ranking is based on Compatible score

Web Service Operations Compatible Scores Calculation

- The SSE scores the candidate operations depending on
  - How well the inputs of the candidate operation can be fed using the outputs of the operation(s) in the workflow.
    - **Input-Output Compatibility Score S_{io}**
  - How well the desired functionality if supplied by the user aligns with the ’ Objective specification˜ of the operation.
    - **Objective Specification Compliance Score S_{obj}**



Ontology

Workflow

WS

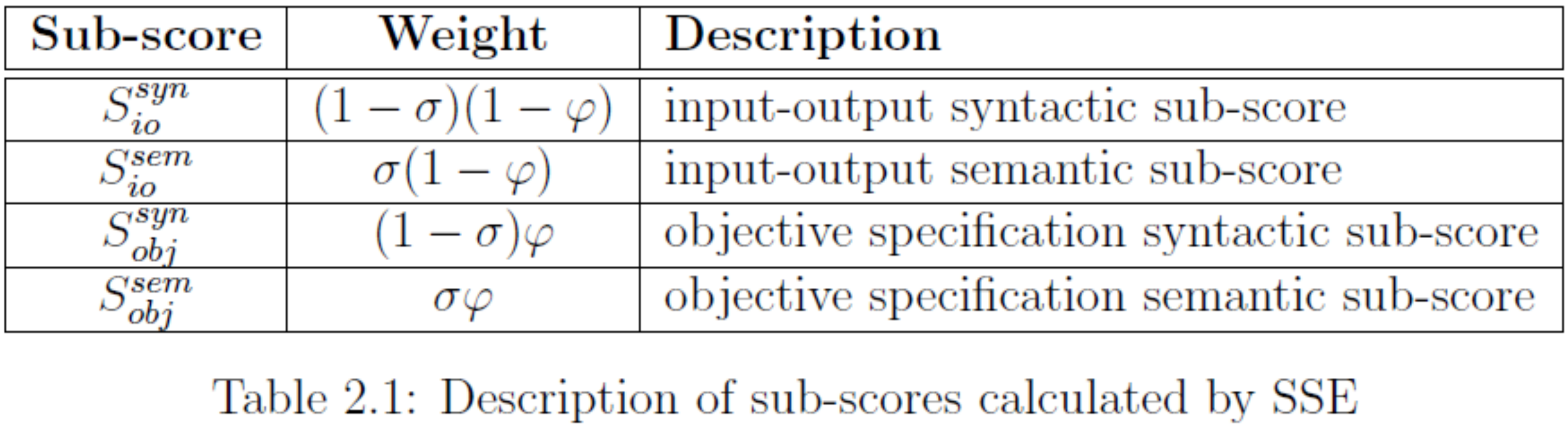
?

<SAWSDL
>

Service Suggestion Engine

Candidate *Web services*

SSE Sub-scores



Sub-score	Weight	Description
S_{io}^{syn}	$(1-\sigma)(1-\varphi)$	input-output syntactic sub-score
S_{io}^{sem}	$\sigma(1-\varphi)$	input-output semantic sub-score
S_{obj}^{syn}	$(1-\sigma)\varphi$	objective specification syntactic sub-score
S_{obj}^{sem}	$\sigma\varphi$	objective specification semantic sub-score

Table 2.1: Description of sub-scores calculated by SSE

σ:	weight of objective specification
(1- σ);	weight of input-output
ψ:	weight of semantic
(1- ψ):	weight of syntactic

Sum of weights = 1

- When calculating these sub-scores, the system calls a ’`Concept similarity`’ module to find out how similar two concepts are.

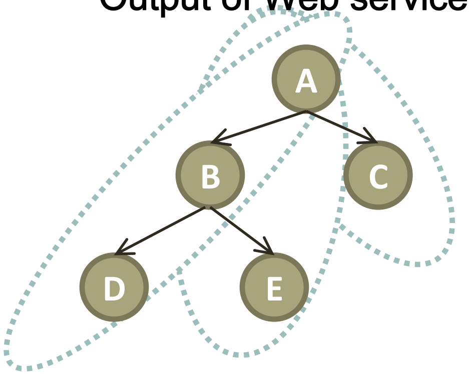
R. Wang, *et al*.Ranking-Based Suggestion Algorithms for Semantic Web Service Composition, ICWS 2010, *pp. 606-613*

Input-Output Compatibility

- Model inputs/outputs using a Directed Acyclic Graph (DAG)
- Transforms the I-O matching problem into a graph pattern matching problem
- SSE supports two algorithms to calculate s_{io} sub-score
  - path based mapping algorithm
  - p-Homomorphism matching algorithm

Path Based Mapping Algorithm

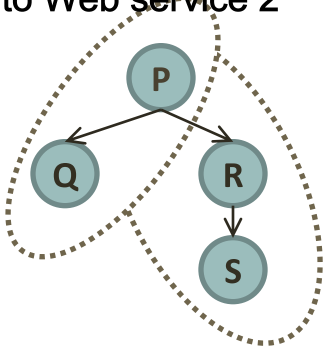
Output of Web service 1



Identify Paths

1. A α̃ B α̃ D
2. A α̃ B α̃ E
3. A α̃ C

Input to Web service 2



Identify Paths

1A̩ P α̃ Q

2A̩ P α̃ R α̃ S

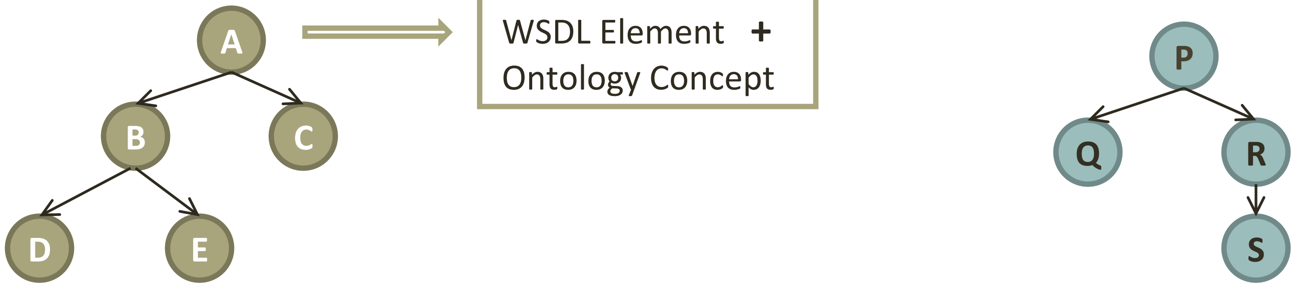
- The goal is to **match** the **input paths** of Web service 1 to **output paths** of Web service 2.
- So in this case we need to find best matching path for

  1’ *P – Q*
  2’ *P – R – S*

Path Based Mapping Algorithm

Output of Web service 1

Input to Web service 2

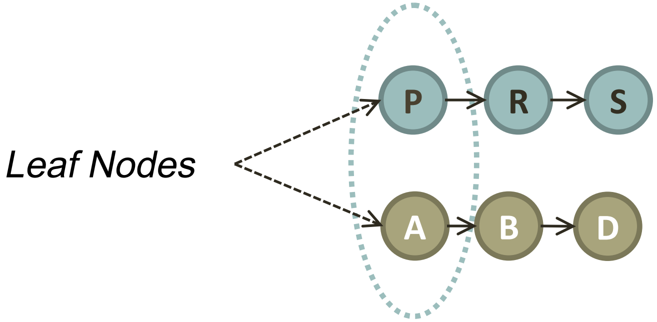


Matching 2Ạ ã 1 P ụ R ụ S
A ụ B ụ D

Finding best matching path for
2Ạ P ụ R ụ S

Comparing paths

- **2’ – 1** P – R – S
  A – B – D
- 2’ – 2 P – R – S
  A – B – E
- 2’ – 3 P – R – S
  A – C



$$\mathrm{w}_1 * SyntacticSimilarity(P,A) + w_2 \ * SemanticSimilarity\ (P,A)$$

$$\mathrm{w}_1 * SyntacticSimilarity(R,B) + w_2 \ * SemanticSimilarity\ (R,B)$$

$$\mathrm{w}_1 * SyntacticSimilarity(S,D) + w_2 \ * SemanticSimilarity\ (S,D)$$

Score for MatchPath (2Ạ ã 1)

Best Matching Path: $\max\{\ MatchPath(2'-1),\ MatchPath(2'-2), MatchPath(2'-3)\ \}$

Path Based Mapping Algorithm

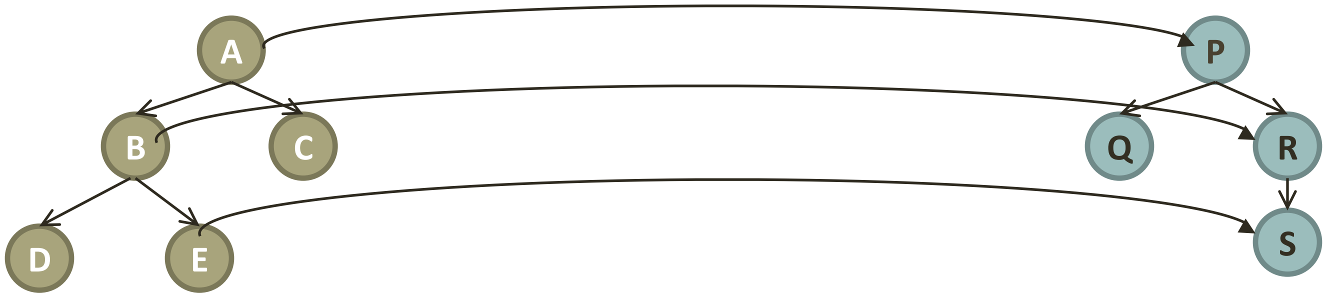
- The final Data mapping score would be weighted sum of best matching paths for all the paths in the input.
- The sub scores are calculated as follows:
- $SyntacticSimilarity(P, A)$: is computed using various string matching algorithms
- $SemanticSimilarity\ (P, A)$: is computed as Concept Similarity between concepts that are used to annotated both the node, $ConceptSimilarity\ (Concept_p, Concept_a)$
- The Concept similarity has been developed as an independent module that computes similarity score between two concepts in the ontology that considers
  - Textual Definitions in the Ontology
  - Logical Definitions in the Ontology (Properties and Restrictions)
  - The hierarchical position of the two concepts in the Ontology

P-Homomorphism Input-Output Matching

- Path-Based matching decomposes the input-output DAGs into individual paths thus losing some structural Information.

- A homomorphism is a structure-preserving map between two algebraic structures (such as groups, rings, or vector spaces)
- Finding an exact Homomorphism mapping between inputs and output DAGs is remote
  - Important to consider similarity between the vertices when considering the mapping

P-Homomorphism Input-Output Matching



Input DAG

Output DAG

Good Matches
A -> P
B -> R
C ->
D -> Q
E -> S

Threshold 0.7

	P	Q	R	S
A	0.8	0.2	0.1	0.5
B	0.4	0.4	0.7	0.3
C	0.2	0.1	0.3	

$$(1-\varphi)\left(\sigma\ SemSim(concept_u, concept_{u'}) + (1-\sigma)\ SynSim(label_u, label_{u'})\right)$$

Objective Specification Compliance

- User can provide the desired functionality as:
  - keywords or
  - a concept in the ontology that he feels closely describes the functionality desired

  - If semantics are available from the user as well as Annotation:
    - *ConceptSimilarity(* ***ObjectiveSpecification*****,** ***DesiredFunctionality*** *)*

  - In absence of semantics from the user:
    - *SyntacticSimilarity (****ConceptDefinition*****,** ***Keywords*** *)*

  - In absence of semantics from the user as well as annotation
    - *SyntacticSimilarity (****OperationName*****,** ***Keywords*** *)*

Workflow Management System

Two popular tools that provide a GUI for creating workflows

- **Galaxy**
  - **easy to use, open-source, Web-based platform that provides multiple tools for bioinformatics data analysis.**
  - **provides an easy way to construct workflows using existing tools in a very simple fashion using a Yahoo pipes-based graphical designer**
  - **Previous work allows adding WebServices as tools to Galaxy**
- Taverna
  - open source
  - is integrated with BioCatalogue, and supports the invocation of web services and their use in workflows.

Evaluation Scenario

ñ Find out more information about a protein sequence and its evolutionary relationships to other protein sequences.

ñ The user might be aware that he wants to

  ñfirst **search a database for similar sequences,**
  ñthen perform **multiple sequence alignment** and
  ñfinally perform **phylogenetic analysis** to construct phylogenetic trees.

ñ Web services already exist for each of the above.

ñ We utilize semantically annotated versions of their descriptions for our example.

Evaluation Set up

- We have evaluated SSE for Suggestions provided for each step against a ranking by a human expert.
- When suggesting from **101 Web service Operations**

Precision:

$$P = \frac{(RelevantResults) \cap (RetrievedResults)}{RetrievedResults}$$

Recall:

$$R = \frac{(RelevantResults) \cap (RetrievedResults)}{RelevantResults}$$

F-Measure:

$$F_\beta = (1+\beta^2)\frac{precision * recall}{\beta^2 \; precision + recall}$$

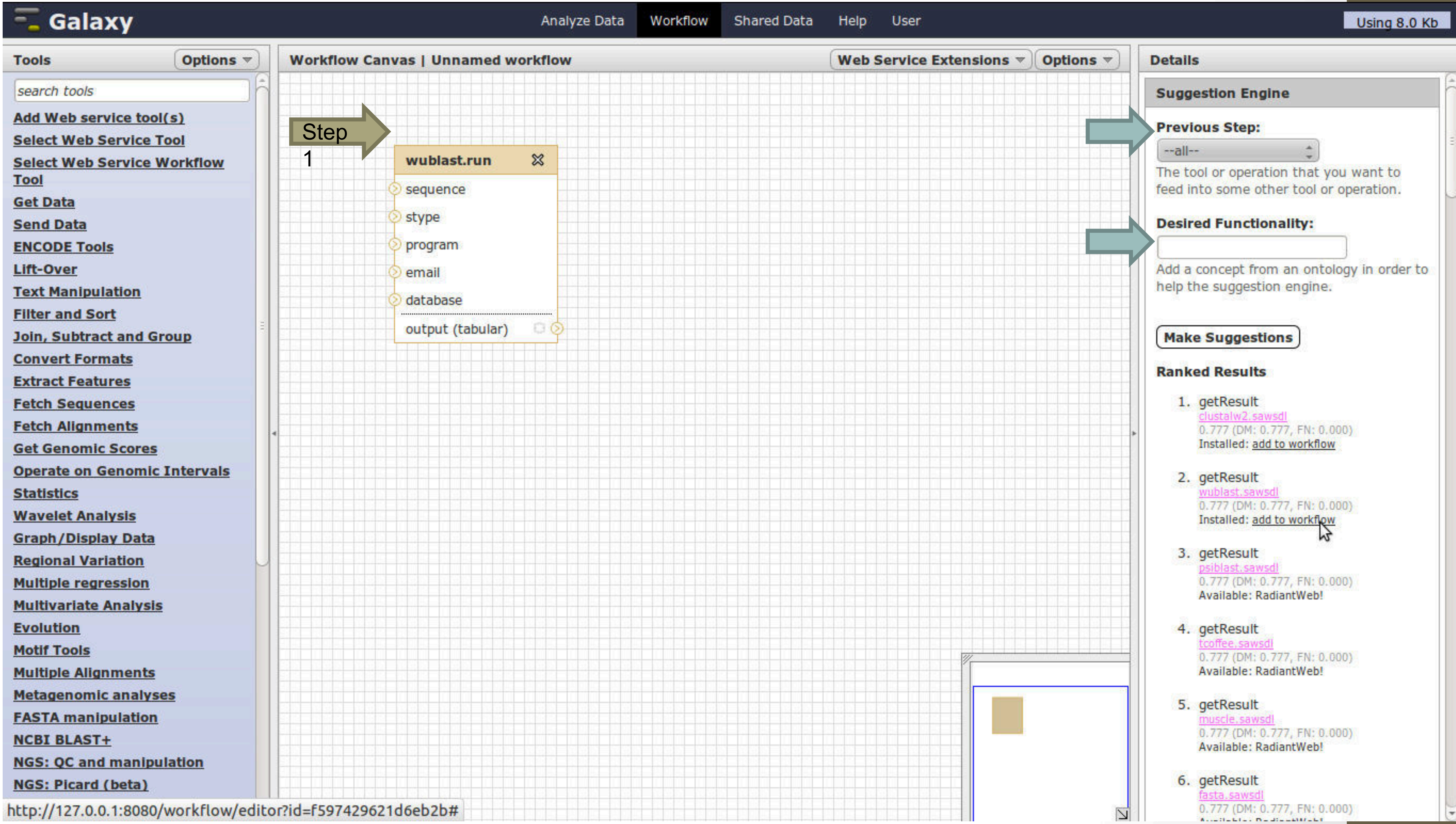
Evaluation: Adding Web Services to Galaxy

- Screen shot

Output

Output

Evaluation: Workflow Construction



Tools Options

search tools

- Add Web service tool(s)
- Select Web Service Tool
- Select Web Service Workflow Tool
- Get Data
- Send Data
- ENCODE Tools
- Lift-Over
- Text Manipulation
- Filter and Sort
- Join, Subtract and Group
- Convert Formats
- Extract Features
- Fetch Sequences
- Fetch Alignments
- Get Genomic Scores
- Operate on Genomic Intervals
- Statistics
- Wavelet Analysis
- Graph/Display Data
- Regional Variation
- Multiple regression
- Multivariate Analysis
- Evolution
- Motif Tools
- Multiple Alignments
- Metagenomic analyses
- FASTA manipulation
- NCBI BLAST+
- NGS: QC and manipulation
- NGS: Picard (beta)

Workflow Canvas | Unnamed workflow Web Service Extensions Options

Step 1

wublast.run

- sequence
- stype
- program
- email
- database
- output (tabular)

http://127.0.0.1:8080/workflow/editor?id=f597429621d6eb2b#

Details

Suggestion Engine

Previous Step: --all--

The tool or operation that you want to feed into some other tool or operation.

Desired Functionality:

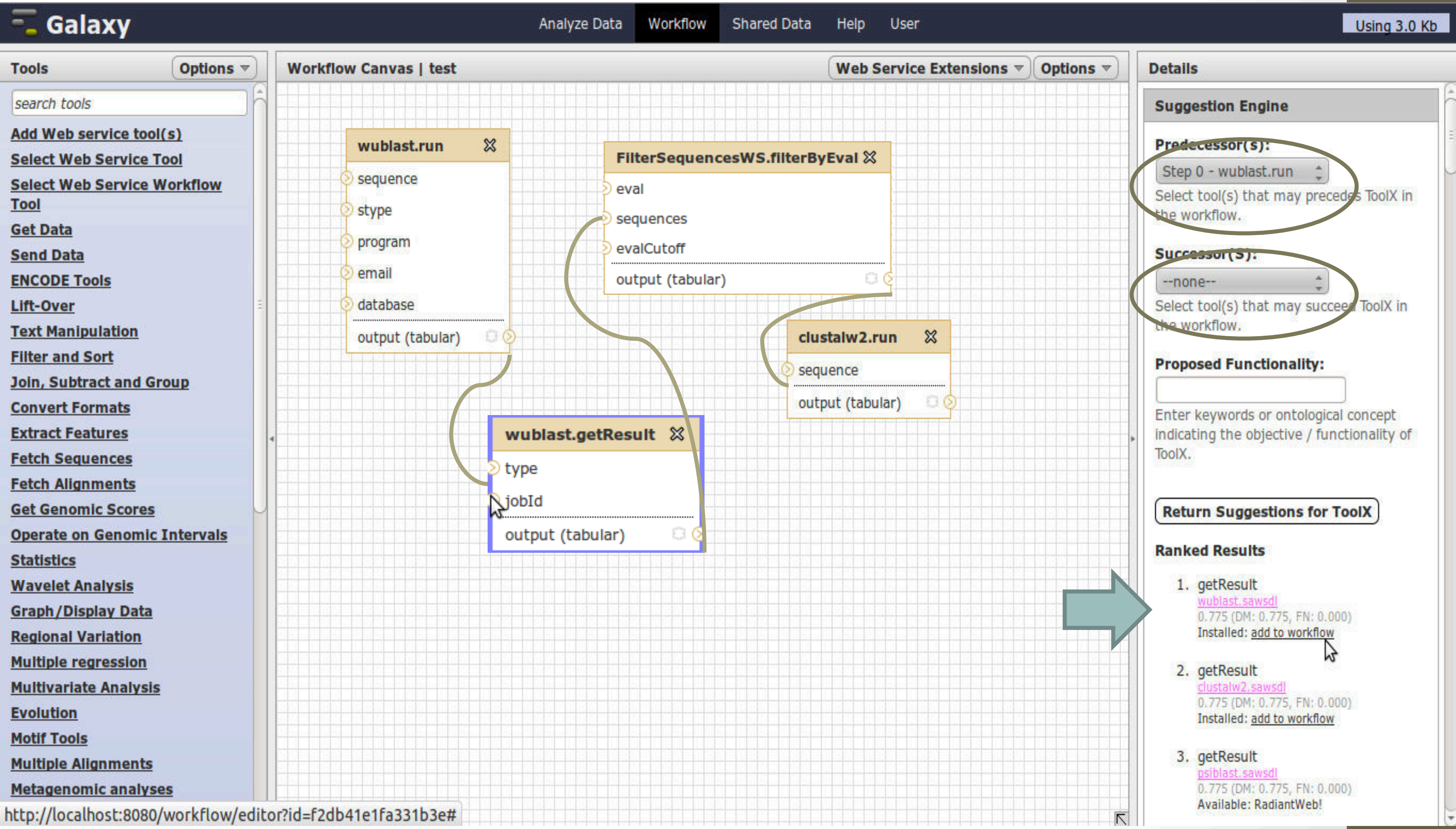
Add a concept from an ontology in order to help the suggestion engine.

Make Suggestions

Ranked Results

1. getResult
   clustalw2.sawsdl
   0.777 (DM: 0.777, FN: 0.000)
   Installed: add to workflow
2. getResult
   wublast.sawsdl
   0.777 (DM: 0.777, FN: 0.000)
   Installed: add to workflow
3. getResult
   psiblast.sawsdl
   0.777 (DM: 0.777, FN: 0.000)
   Available: RadiantWeb!
4. getResult
   tcoffee.sawsdl
   0.777 (DM: 0.777, FN: 0.000)
   Available: RadiantWeb!
5. getResult
   muscle.sawsdl
   0.777 (DM: 0.777, FN: 0.000)
   Available: RadiantWeb!
6. getResult
   fasta.sawsdl
   0.777 (DM: 0.777, FN: 0.000)

Evaluation: Extend Workflow



Suggestion Engine

Predecessor(s): Step 0 - wublast.run

Select tool(s) that may precedes ToolX in the workflow.

Successor(s): --none--

Select tool(s) that may succeed ToolX in the workflow.

Proposed Functionality:

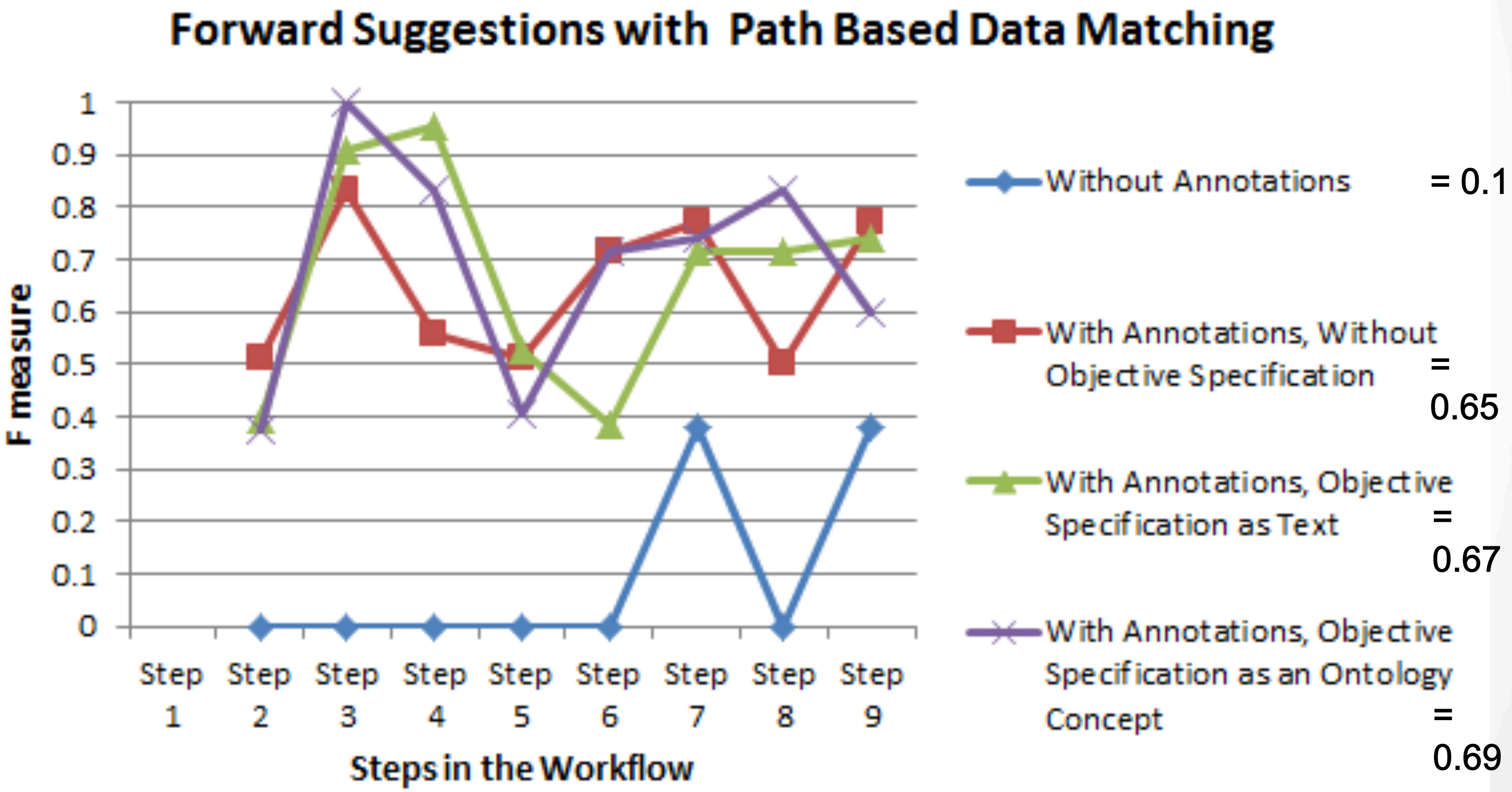
Enter keywords or ontological concept indicating the objective / functionality of ToolX.

Ranked Results

1. getResult
   wublast.sawsdl
   0.775 (DM: 0.775, FN: 0.000)
   Installed: add to workflow
2. getResult
   clustalw2.sawsdl
   0.775 (DM: 0.775, FN: 0.000)
   Installed: add to workflow
3. getResult
   psiblast.sawsdl
   0.775 (DM: 0.775, FN: 0.000)
   Available: RadiantWeb!

http://localhost:8080/workflow/editor?id=f2db41e1fa331b3e#

Evaluation: Forward Suggestions (Path Based)



= 0.1

= 0.65

= 0.67

= 0.69

Conclusions & Future Work

- The use case demonstrated,
  - that of choosing from 101 operations for a 9-step workflow, presents a challenging task for a user (without a tool support).
- Semantics from ontology can definitely help in different aspects of Web service Compositions.
- SSE can help the user considerably narrow down the choices for the next or previous step.
- The availability of a SSE as a Web service will facilitate easier integration with existing workflow composition tools.
- **Future work**
  - Other kind of web services, like secondary structure analysis, sequence annotation web services, etc.
  - plug-in for other workflow editors, *e.g.*, Taverna
- Code and files available at:
  - http://mango.ctegd.uga.edu/jkissingLab/SWS/index.html

Acknowledgements

Alok Dhamanaskar: alokdhamanaskar@gmail.com

University of Georgia

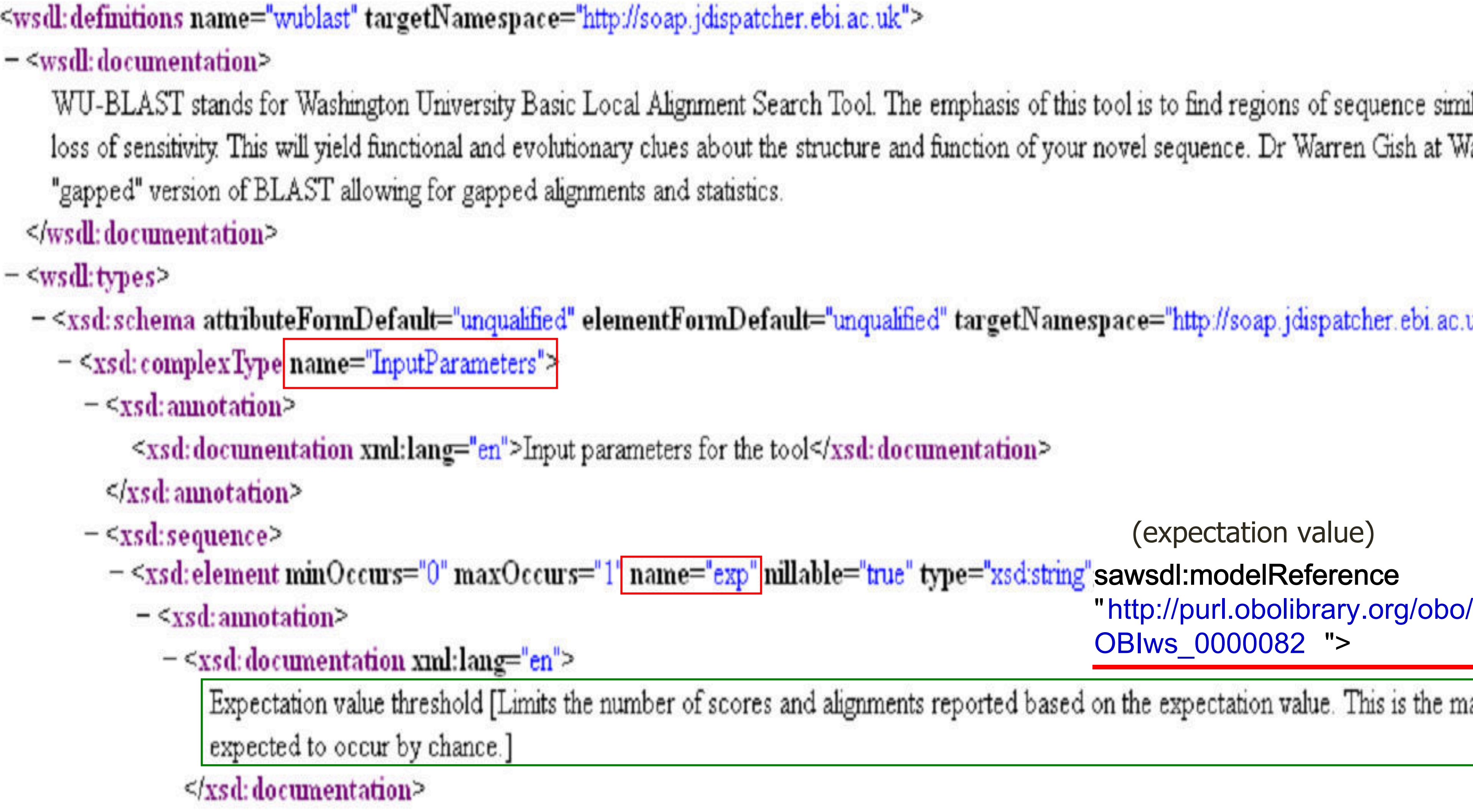
- Michael E. Cotterell
- Jessica C. Kissinger
- John Miller

University of Pennsylvania

- Jie Zheng
- Christian J. Stoeckert, Jr.

Funding: NIH R01 GM093132

Adding ontology annotation into WSDL / WADL files



```
<wsdl:definitions name="wublast" targetNamespace="http://soap.jdispatcher.ebi.ac.uk">
- <wsdl:documentation>
    WU-BLAST stands for Washington University Basic Local Alignment Search Tool. The emphasis of this tool is to find regions of sequence simi
    loss of sensitivity. This will yield functional and evolutionary clues about the structure and function of your novel sequence. Dr Warren Gish at Wa
    "gapped" version of BLAST allowing for gapped alignments and statistics.
  </wsdl:documentation>
- <wsdl:types>
  - <xsd:schema attributeFormDefault="unqualified" elementFormDefault="unqualified" targetNamespace="http://soap.jdispatcher.ebi.ac.u
    - <xsd:complexType name="InputParameters">
      - <xsd:annotation>
          <xsd:documentation xml:lang="en">Input parameters for the tool</xsd:documentation>
        </xsd:annotation>
      - <xsd:sequence>
        - <xsd:element minOccurs="0" maxOccurs="1" name="exp" nillable="true" type="xsd:string"
          - <xsd:annotation>
            - <xsd:documentation xml:lang="en">
                Expectation value threshold [Limits the number of scores and alignments reported based on the expectation value. This is the ma
                expected to occur by chance.]
              </xsd:documentation>
```

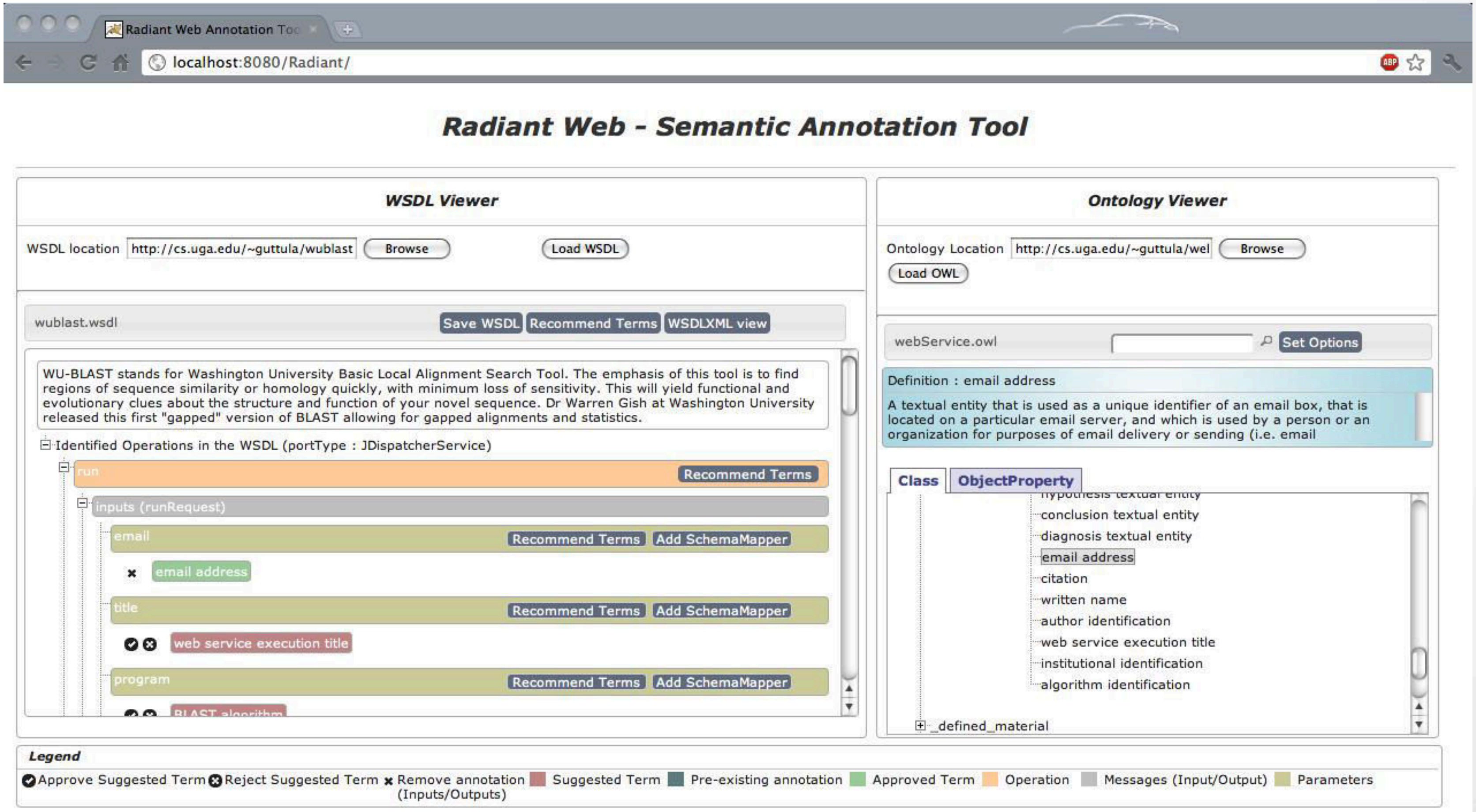
(expectation value)

sawsdl:modelReference
"http://purl.obolibrary.org/obo/
OBIws_0000082 ">

RadiantWeb Annotation Tool

ñ Manually annotation is labor intensive and error prone

ñ RadiantWeb ã suggest ontology terms, generate SAWSDL file



Radiant Web Annotation Too

localhost:8080/Radiant/

Radiant Web - Semantic Annotation Tool

WSDL Viewer

WSDL location: http://cs.uga.edu/~guttula/wublast

wublast.wsdl

WU-BLAST stands for Washington University Basic Local Alignment Search Tool. The emphasis of this tool is to find regions of sequence similarity or homology quickly, with minimum loss of sensitivity. This will yield functional and evolutionary clues about the structure and function of your novel sequence. Dr Warren Gish at Washington University released this first "gapped" version of BLAST allowing for gapped alignments and statistics.

⊟ Identified Operations in the WSDL (portType : JDispatcherService)

- run — Recommend Terms
  - inputs (runRequest)
    - email — Recommend Terms | Add SchemaMapper
      - ✖ email address
    - title — Recommend Terms | Add SchemaMapper
      - ✓ ⊗ web service execution title
    - program — Recommend Terms | Add SchemaMapper
      - ✓ ⊗ BLAST algorithm

Ontology Viewer

Ontology Location: http://cs.uga.edu/~guttula/wel

webService.owl

Definition : email address

A textual entity that is used as a unique identifier of an email box, that is located on a particular email server, and which is used by a person or an organization for purposes of email delivery or sending (i.e. email

- hypothesis textual entity
- conclusion textual entity
- diagnosis textual entity
- email address
- citation
- written name
- author identification
- web service execution title
- institutional identification
- algorithm identification

⊞ _defined_material

Legend

✓ Approve Suggested Term ⊗ Reject Suggested Term ✖ Remove annotation (Inputs/Outputs) 🟥 Suggested Term ■ Pre-existing annotation 🟩 Approved Term 🟧 Operation ■ Messages (Input/Output) ■ Parameters