IPAW 2012
International Provenance and Annotation Workshop
Santa Barbara, California, June 19-22

Using Domain-Specific Data to Enhance Scientific Workflow Steering Queries

João Gonçalves[1], Daniel de Oliveira[1], Eduardo Ogasawara[2], Kary Ocaña[1] and Marta Mattoso[1]

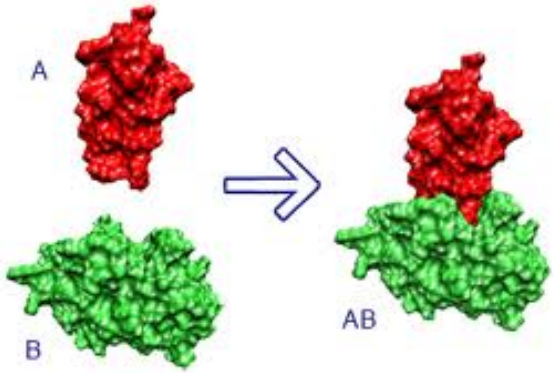
Federal University of Rio de Janeiro
Rio de Janeiro, Brazil[1]
Federal Center of Technological Education
Rio de Janeiro, Brazil[2]

Large scale scientific experiments

- Correspond to simulations, mathematical and computational models
- Process a large amount of data
- Modeled as scientific workflows
- Assisted by scientific workflow management systems (SWfMS)
- Must gather provenance data

Exploratory workflows

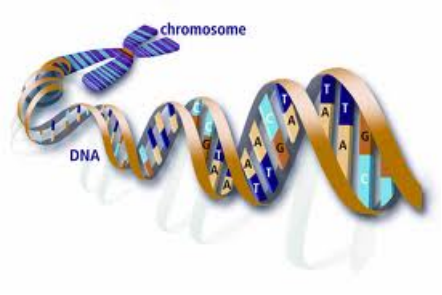
- Scientists have to explore the behavior of their model under different inputs
  - This occurs in many areas such as bioinformatics, computational fluid dynamics, uncertainty quantification, dark energy analysis



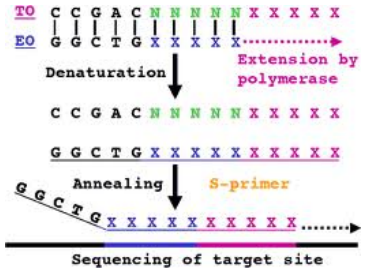
- These data-centric workflows are computationally intensive and they may run for hours/days
- Demand distributed and parallel processing

An example of a large scale experiment: Comparative Genomics workflow

Input data for comparative genomics analysis: DNA and RNA sequences...
(200 sequences)



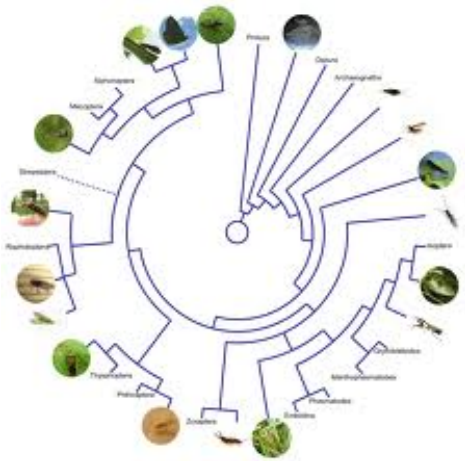
1. Sequence alignment (MAFFT, ProbCons, ClustalW, Muscle and Kalign) and conversion (phylip)



Large volume of data is generated (aligned sequences)...

2. ... which are used to compute a new evolutive model (modelgenerator)...
(304.2 hours)

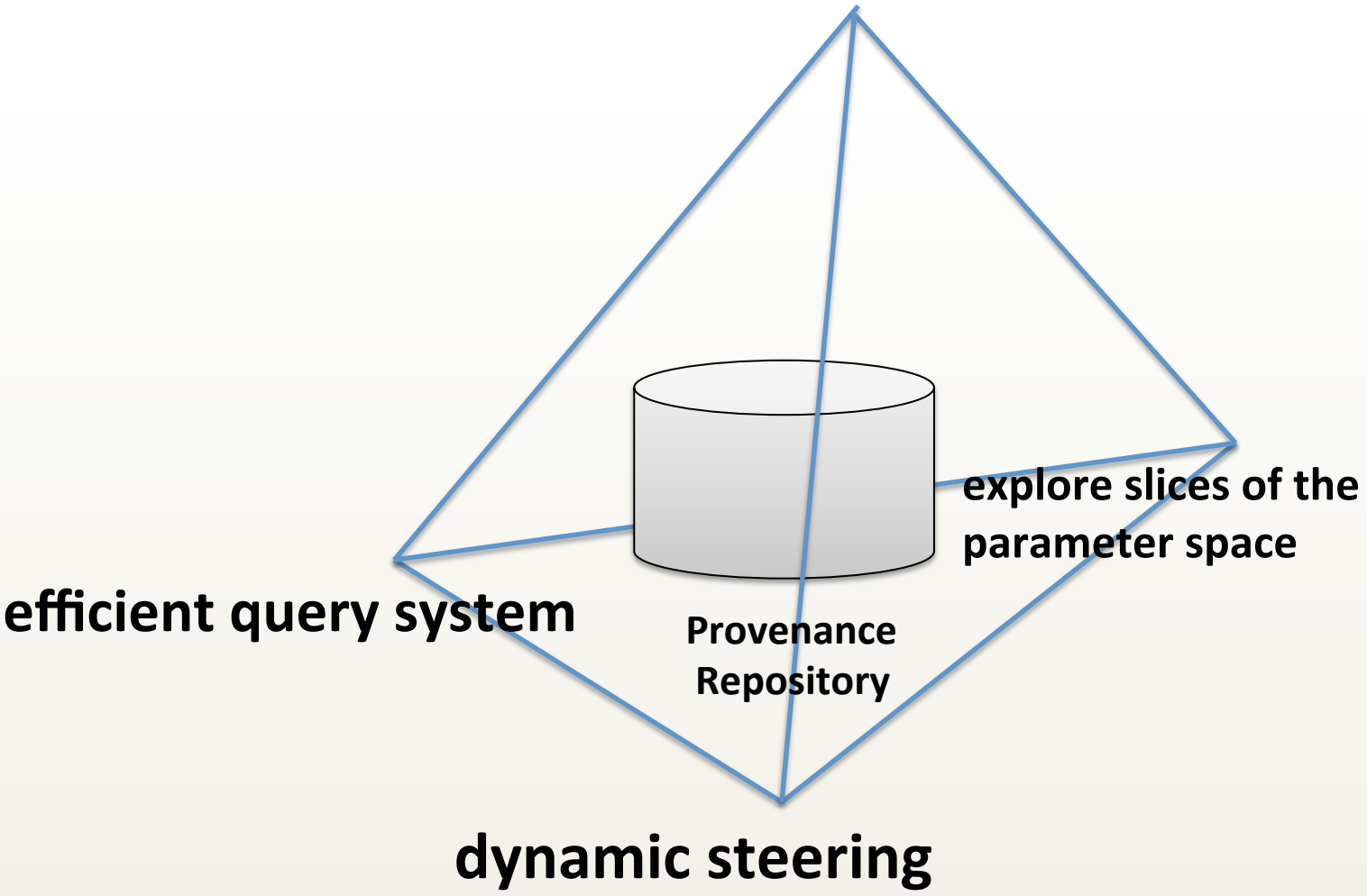
3. ...and to generate phylogenetic trees
(RAxML)



Challenges in exploratory workflows

- Interaction with users during workflow execution
  - Enable scientists to analyze the status of the workflow execution
  - Interfere by stopping or changing the space of parameters to be explored
  - Very important in cloud environments
  - Fundamental technology to fully support e-Science (Gil et al. (2007))

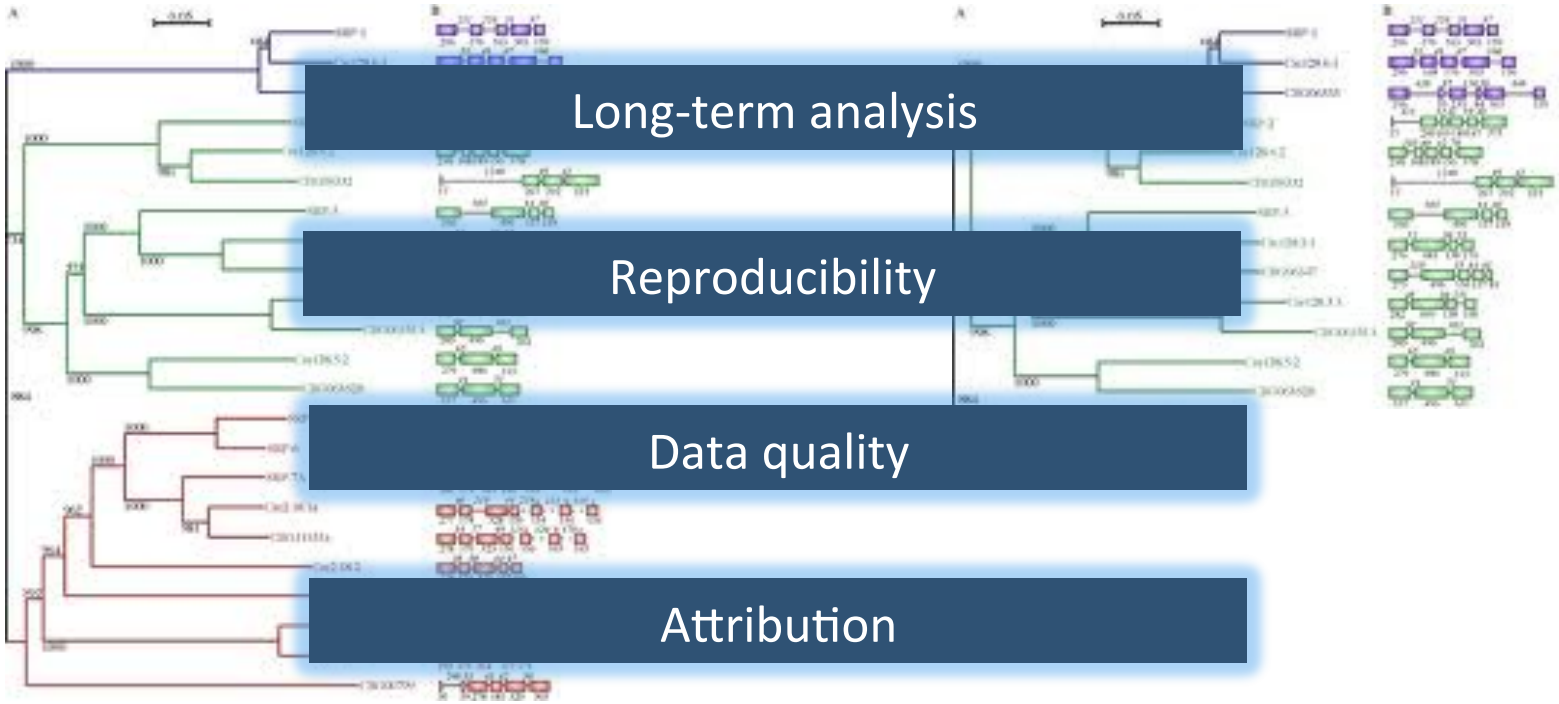
Characteristics in user-steered workflows



Current usage of provenance

iii-201102281EHWP25EX8.tcoffeeStockholm

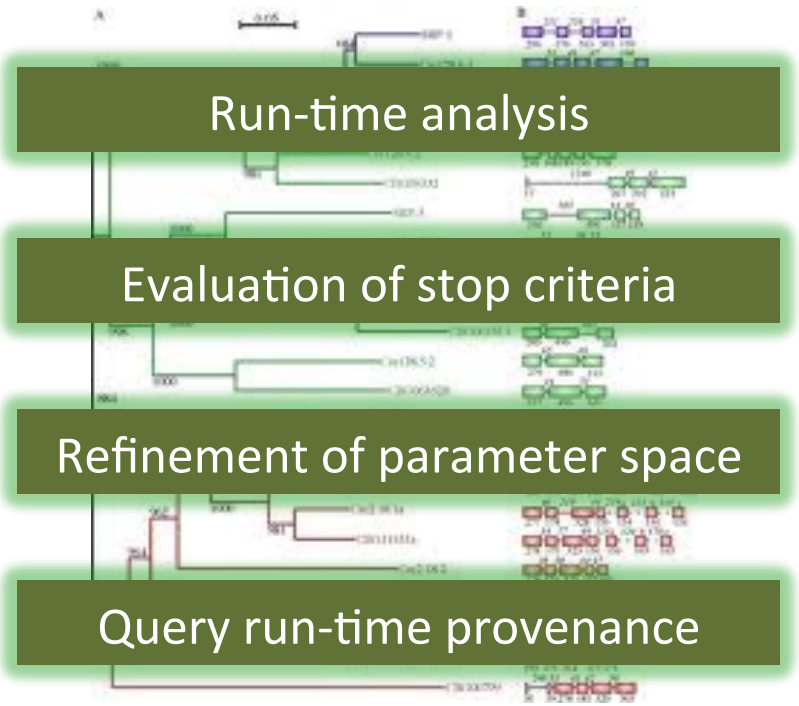
iii-201102281EHWP25EX9.tcoffeeStockholm



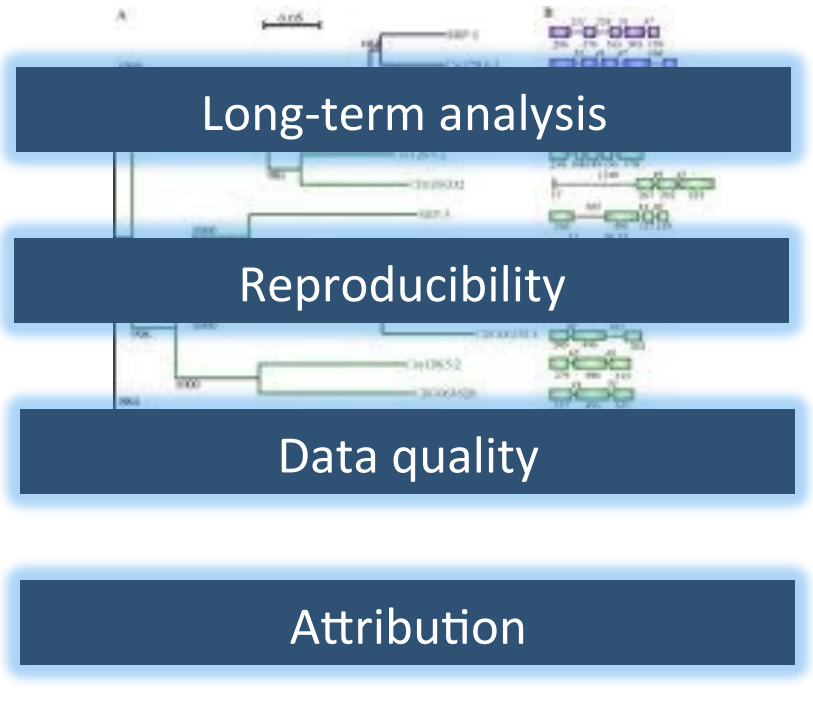
How were these trees created? Who created them?
What is the difference? Are they associated to the same input dataset?

Using provenance for steering

iii-201102281EHWP25EX8.tcoffeeStockholm



iii-201102281EHWP25EX9.tcoffeeStockholm



Is the iteration converging ? Should I change my time step?
Should I skip this branch of the parameter space ? Should I stop some activities?
Should I refine some parameters for better results?
Are there activities returning errors that may demand re-execution?

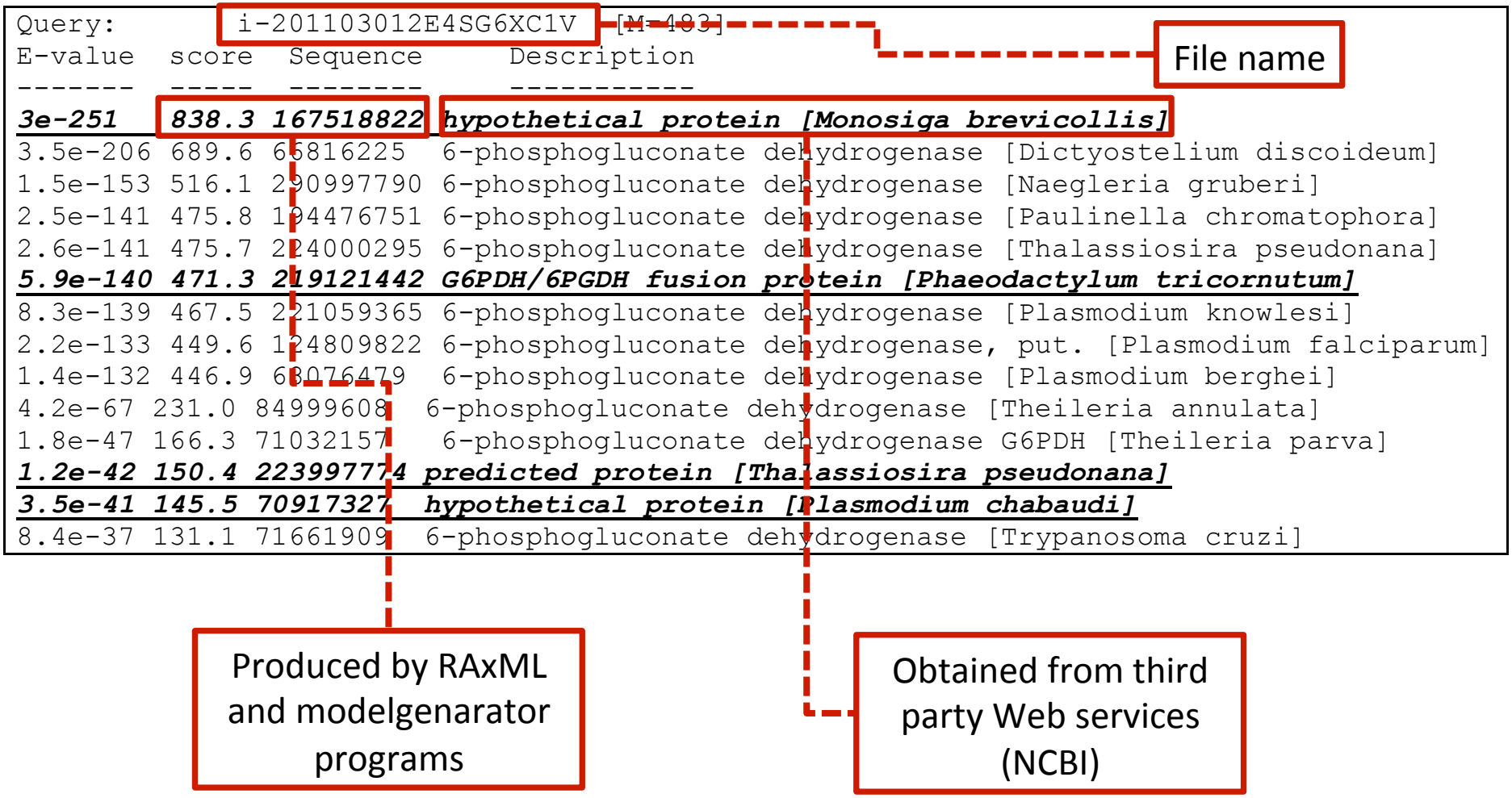
Provenance for runtime analysis

- Runtime provenance data analysis is the basis for workflow steering
  - Derivation history of a data product and the status of all activities should be available as soon as they execute and produce their data

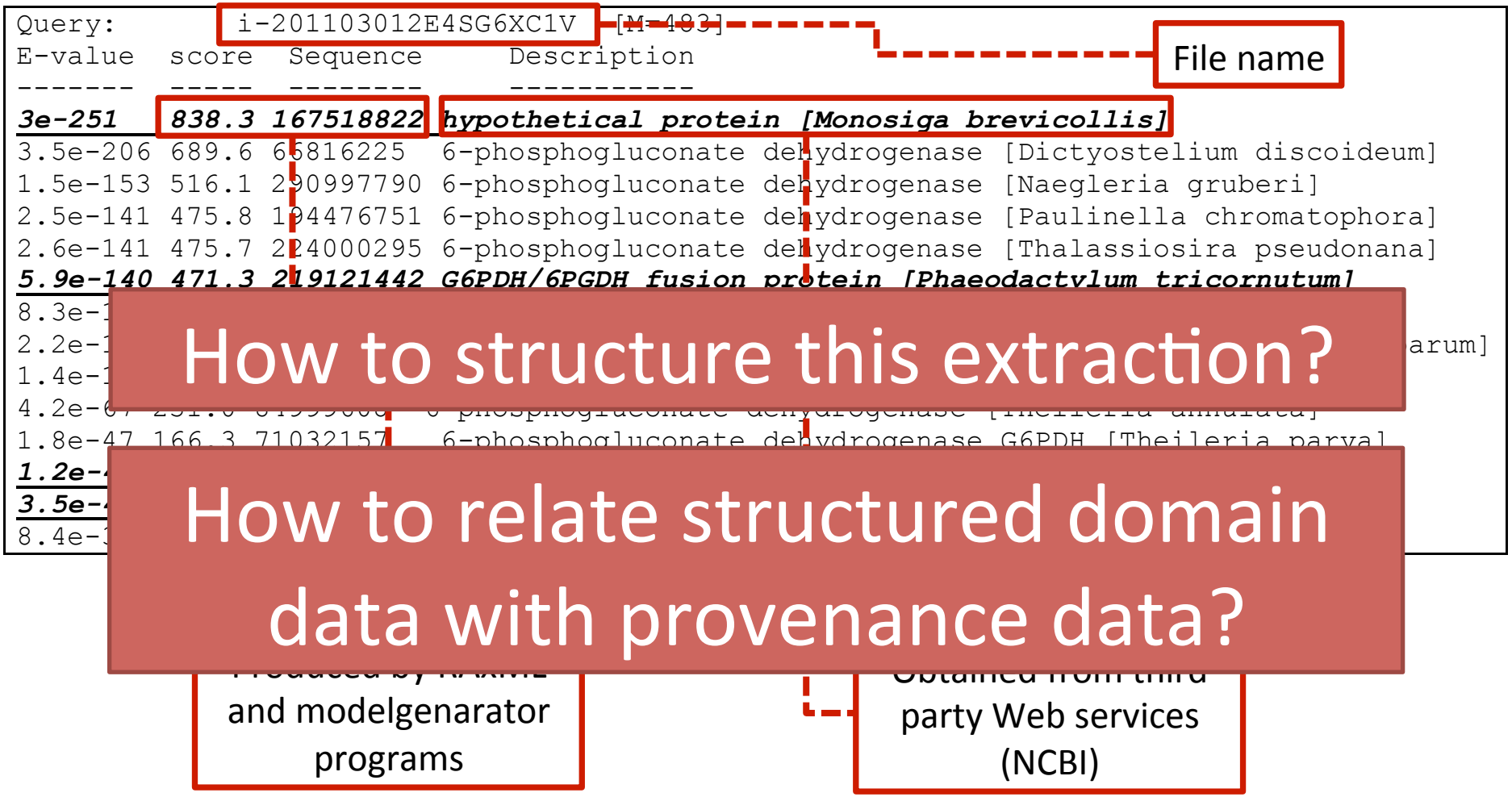
- Provenance alone is not sufficient to support steering mechanisms in workflows

Domain-specific data is also needed!

Extracting Domain-specific Data

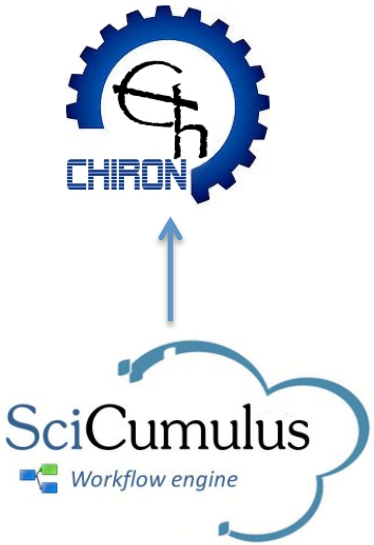


Extracting Domain-specific Data

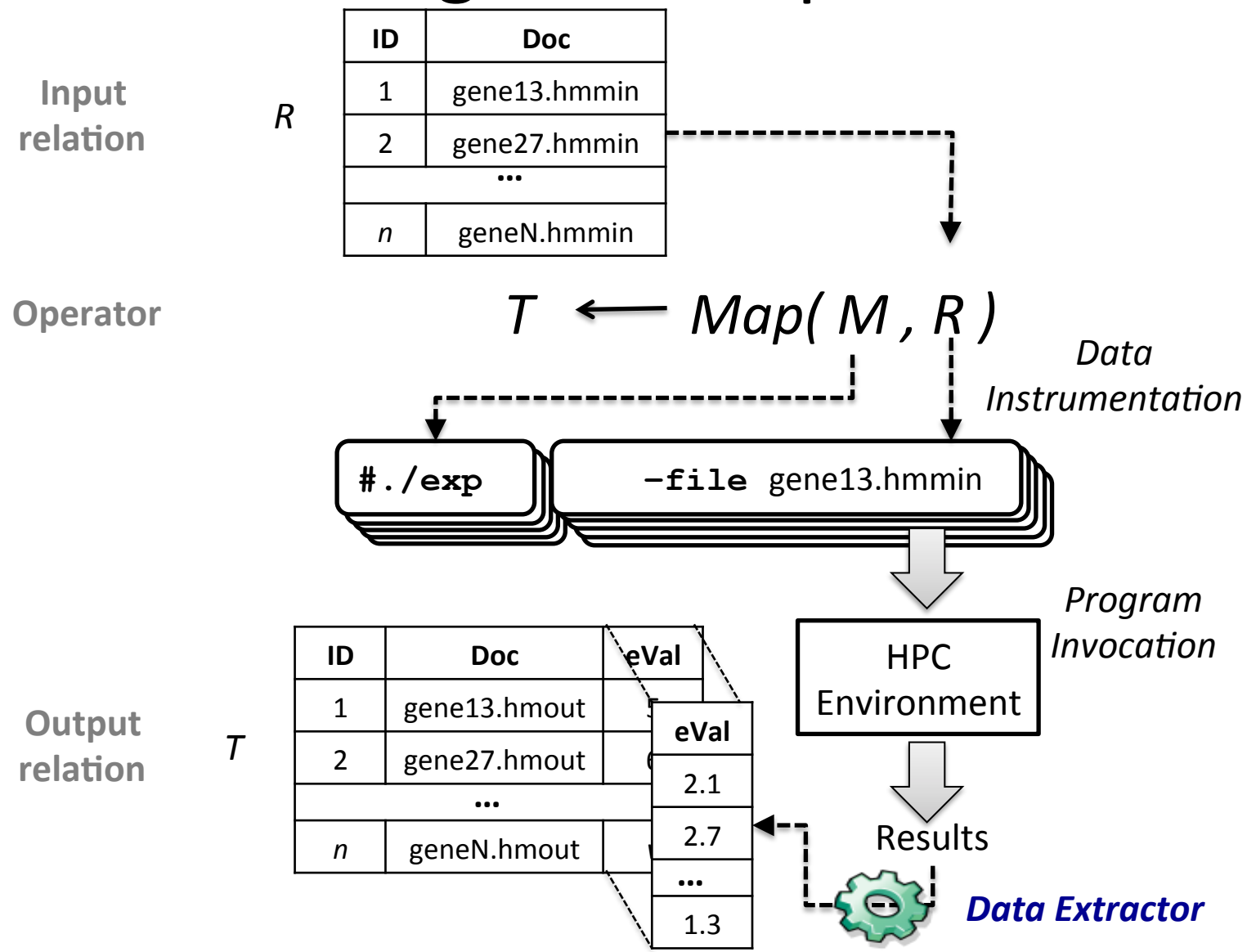


SciCumulus Workflow Engine

- Designed to distribute scientific workflows to execute in cloud environments
- Manages and orchestrates the execution of many activities (data and programs) on a distributed set of virtual machines (VMs)
- Provides provenance data at runtime
- Based on algebraic expressions



Workflow execution based on algebraic expressions



Data Extractor component

- How to extract data using a framework that supports different formats (e.g. binary, Fasta, HDF5, etc.) ?
- Data extractor component (algebra based)
  - Invokes an external program (defined) by scientists that analyzes produced data files and extracts domain-specific data from it
  - Programs encapsulate domain-specific extraction rules
  - Workflow engine provides runtime provenance data
  - Algebraic relations link domain data to provenance data

Using Data Extractor in SciCumulus

```
<SciCumulus>
 <database name="scicumulus" server="mp4-5b.dyndns.info" port="5432"/>
 <SciCumulusWorkflow tag="SciEvol" description="MER" exectag="scievol" expdir="/scievol/">
  <SciCumulusActivity tag="MSA" activation="./experiment.cmd" extractor="./extract_msa.cmd" >
   <Relation reltype="Input" name="rel_in_1" filename="input_step_1.txt"/>
   <Relation reltype="Output" name="rel_out_1" filename="output_step_1.txt"/>
   <File filename="experiment.cmd" instrumented="true"/>
  </SciCumulusActivity>
 </SciCumulusWorkflow>
</SciCumulus>
```

k	file
1	gene13.hmmin

a

./experiment.cmd

k	file
1	gene13.hmmout

b

./extract_msa.cmd

Tp$_1$

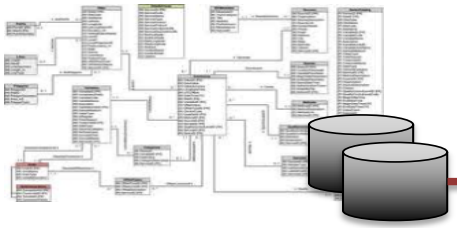
Tp$_2$

k	file	*e-value*	score	fasta_ID	fasta_description	taxonomy	is_valid
1	gene13.hmmout	3e-251	838.3	167518822	6-phosphogluconate dehydrogenase	*Monosiga brevicollis*	True

Tp$_3$

c

./extract_msa.cmd

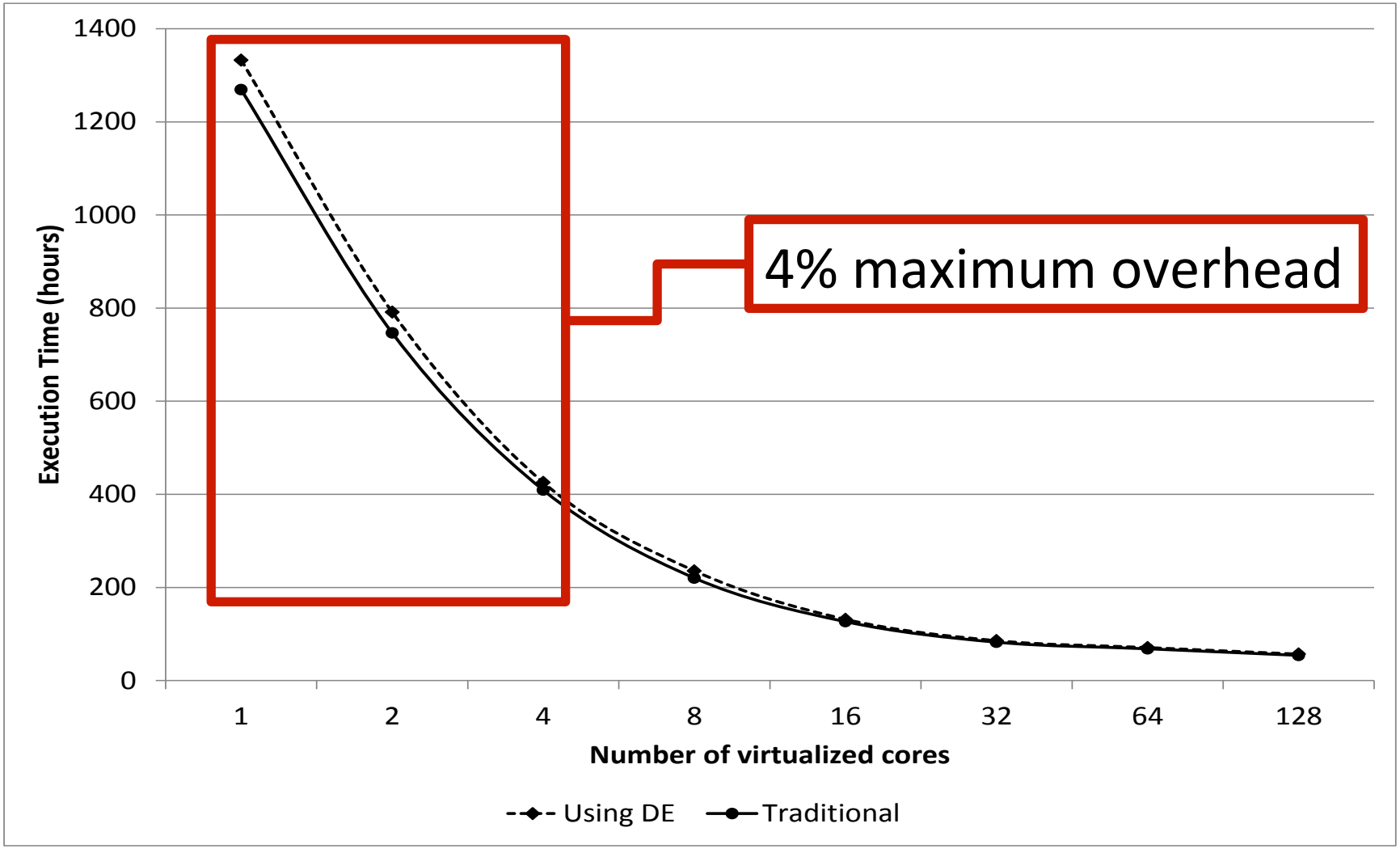


Provenance Repository

Experimental evaluation setup

- Executed in Amazon EC2 environment
- 128 micro-size VMs in the virtual cluster
- Two separate executions of SciHmm were performed
  - SciHmm without inserting DE in the workflow
  - SciHmm with DE, which analyzes the produced set of data and extracts domain-specific information
- Overhead analysis
- Exploration of potentials queries

Experimental evaluation



Run-Time Query #1 using domain data

- Which sequences in available hits do not belong to a specific gene (given as parameter by scientists)

```
SELECT S.NCBI_Ref_Sequence

FROM Task T, Hits H, MSA_CONVERTED M2, MSA M1,
                     SEQUENCE_GROUP SG, SEQUENCE S,
                     ORGANISM O, SPECIE SP, GENUS G

WHERE T.taskid = H.taskid AND H.msacid = M2.msacid
AND M2.msaid = M1.msaid AND M1.sgid = SG.sgid
AND SG.sqid = S.sqid AND S.orgid = O.orgid
AND O.spid = SP.spid and SP.genid = G.genid
AND T.exitStatus = 0 /* No error */
AND G.genus = "PLASMODIUM"
```

Run-Time Query #2 using domain data

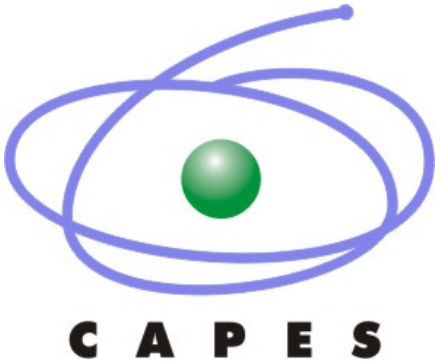
- How many sequences from input data are annotated as putative and hypothetical that denote that sequences are not annotated as “true genes”
  - SELECT COUNT(*)
    FROM Task T, Hits H, MSA_CONVERTED M2, MSA M1, SEQUENCE_GROUP SG, SEQUENCE S
    WHERE T.taskid = H.taskid AND H.msacid = M2.msacid
    AND M2.msaid = M1.msaid AND M1.sgid = SG.sgid
    AND SG.sqid = S.sqid AND T.status = “FINISHED”
    AND T.exitStatus = 0 /* No error */
    AND (S.NCBI_Ref_Sequence LIKE “%PUTATIVE%” OR
    S.NCBI_Ref_Sequence LIKE “%HYPOTHETICAL%” OR
    S.NCBI_Ref_Sequence LIKE “%SIMILAR%”)

Conclusions

- This paper proposes an approach for extracting domain-specific data from produced data files and storing them along with provenance data
- It can be used as a basis for steering mechanisms
- One of the advantages of our solution is modeling domain-specific data in the same formalism of the workflow algebra
- The approach presents small runtime overhead and allows for building complex steering mechanisms based on enriched provenance data with domain-specific data

- As future work, we intend to adopt current efforts in simplifying SQL query interface to users such as in Jagadish et al. (SIGMOD'07) e Gadelha et al. (TaPP'11)

Acknowledgements



Federal University Rio de Janeiro

NACAD
High Performance Computing Center

IPAW 2012
International Provenance and Annotation Workshop
Santa Barbara, California, June 19-22

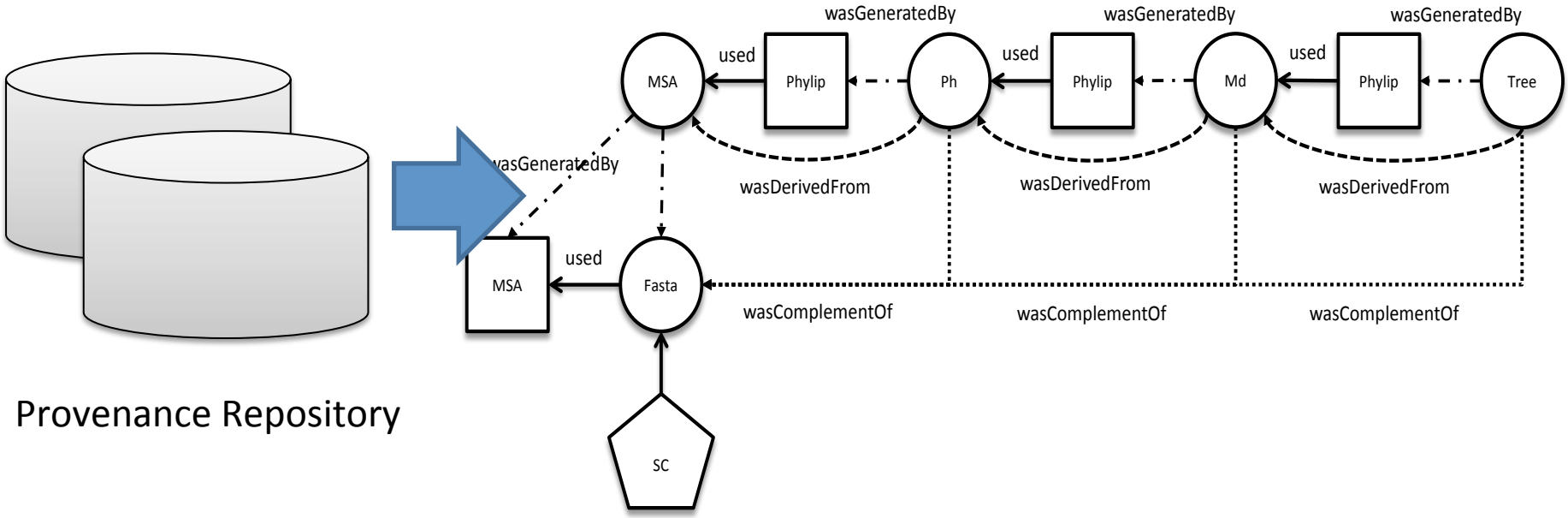
Using Domain-Specific Data to Enhance Scientific Workflow Steering Queries

João Gonçalves[1], Daniel de Oliveira[1], Eduardo Ogasawara[2], Kary Ocaña[1] and Marta Mattoso[1]

Federal University of Rio de Janeiro
Rio de Janeiro, Brazil[1]
Federal Center of Technological Education
Rio de Janeiro, Brazil[2]

OPM compliancy

- The workflow engine generates the OPM wasGeneratedBy between the activity and data file and the domain contents



Related work

- Provenance enriched with domain specific data
  - Karma (2006) – framework for collecting provenance with domain specific data in heterogeneous environment
  - Provenir (2009) – upper level ontology for provenance representation
  - Janus (2010) – extension of Provenir for modeling domain specific data
- Support for complex queries
  - Anad et al (2009) – support for domain specific queries in nested collections
  - Gadelha et al (2011) – query interface for building SQL queries for Swift
- None of approaches support steering queries mixing runtime-provenance with domain-specific data