Tavaxy

Integrating Taverna and Galaxy with Cloud Computing Support

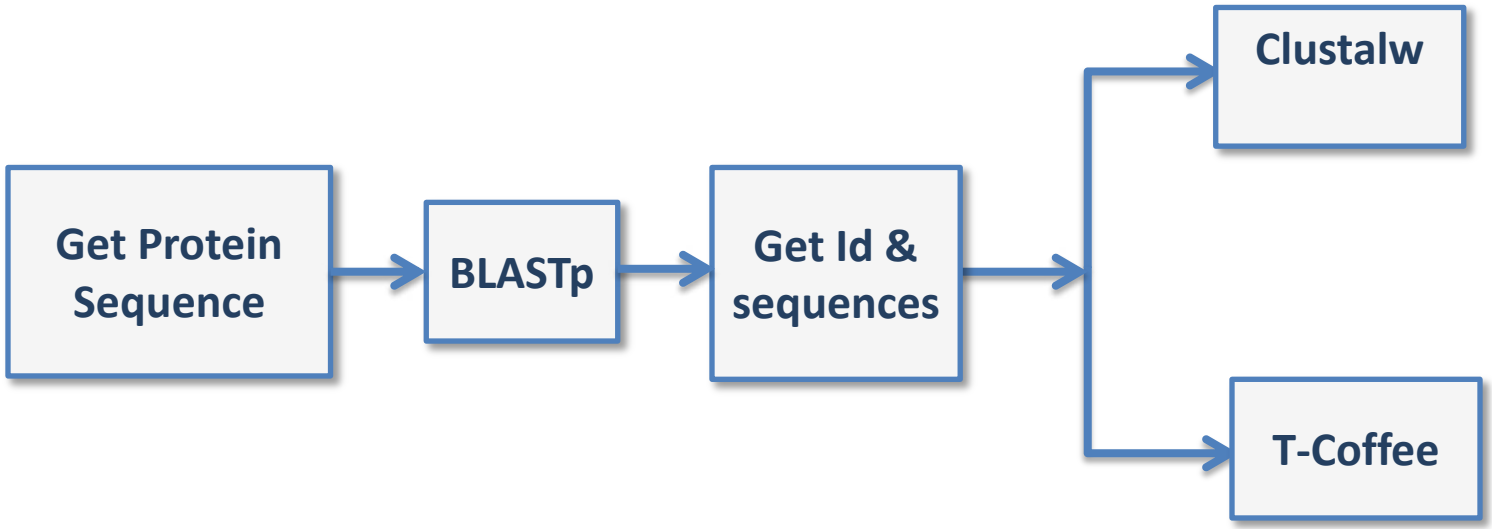
Mohamed Abouelhoda

Nile University
Egypt

Nile University

Workflows in Bioinformatics

Finding Homologous Sequences



Implementing Scientific Workflows

Method 1: Write Python/Perl/Shell script

Advantages

- Reliable and efficient
- Comprehensive programming capabilities (conditionals, loops, etc..)

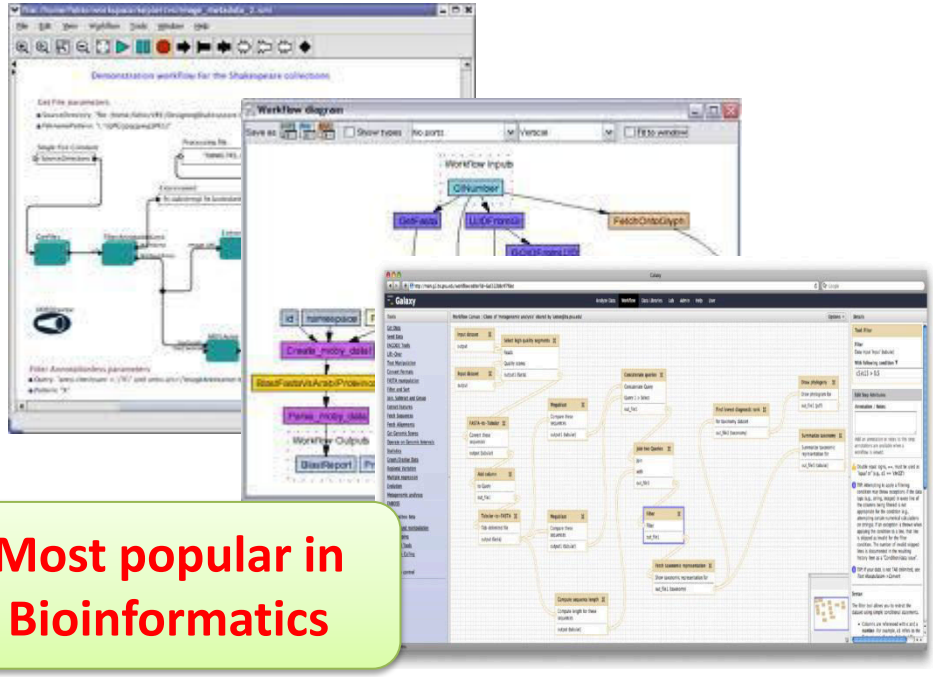
Disadvantages

- Requires programming skills , especially with HPC resources
- Scripts are workflow-specific
- Costly to create, debug and modify
- Requires installing and managing tools

Implementing Scientific Workflows

Methods 2: Use of Workflow Systems

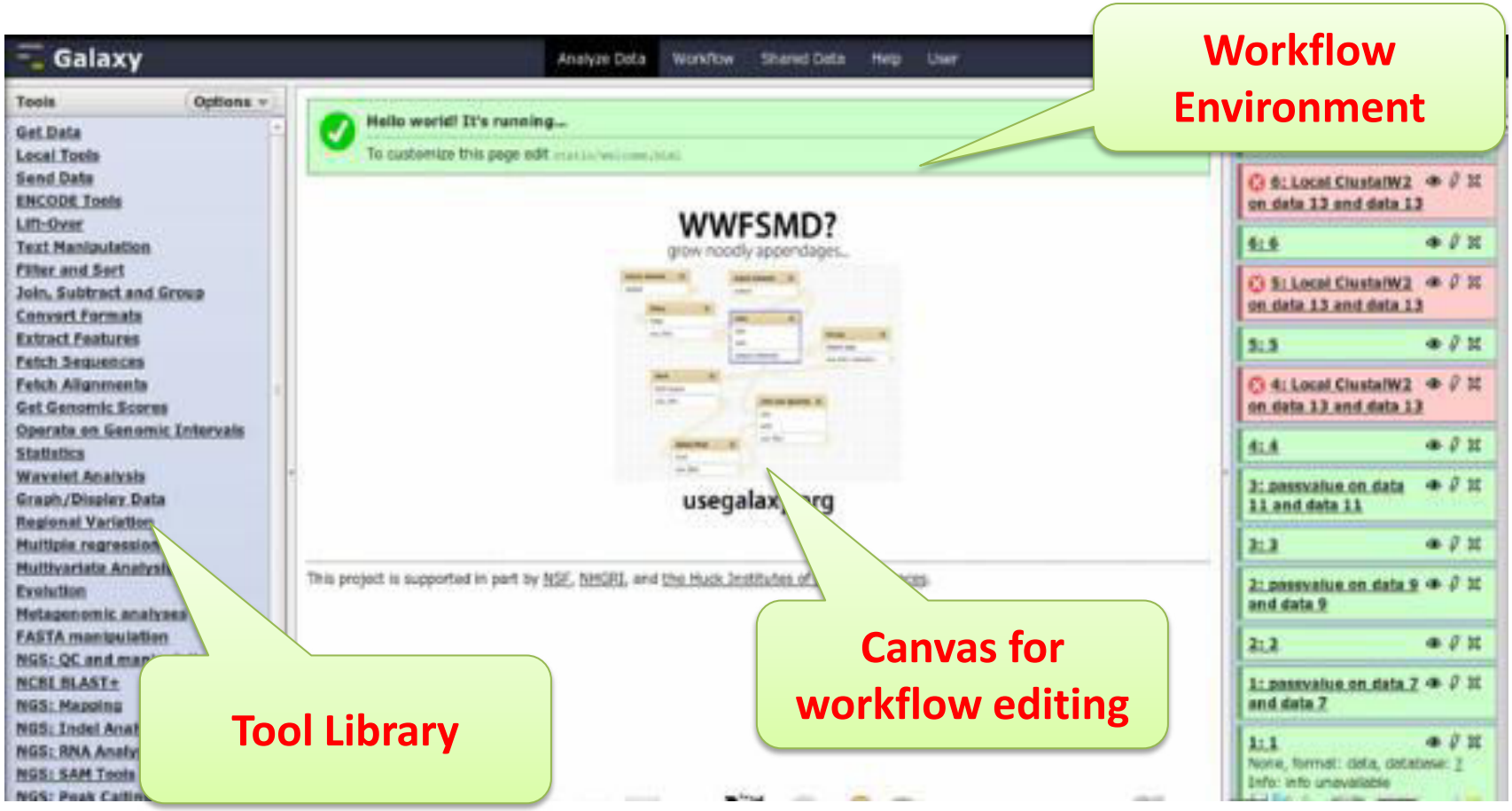
- Kepler
- Triana
- WildFire
- GenePattern
- Pegsus
- Taverna
- Galaxy
- and many more



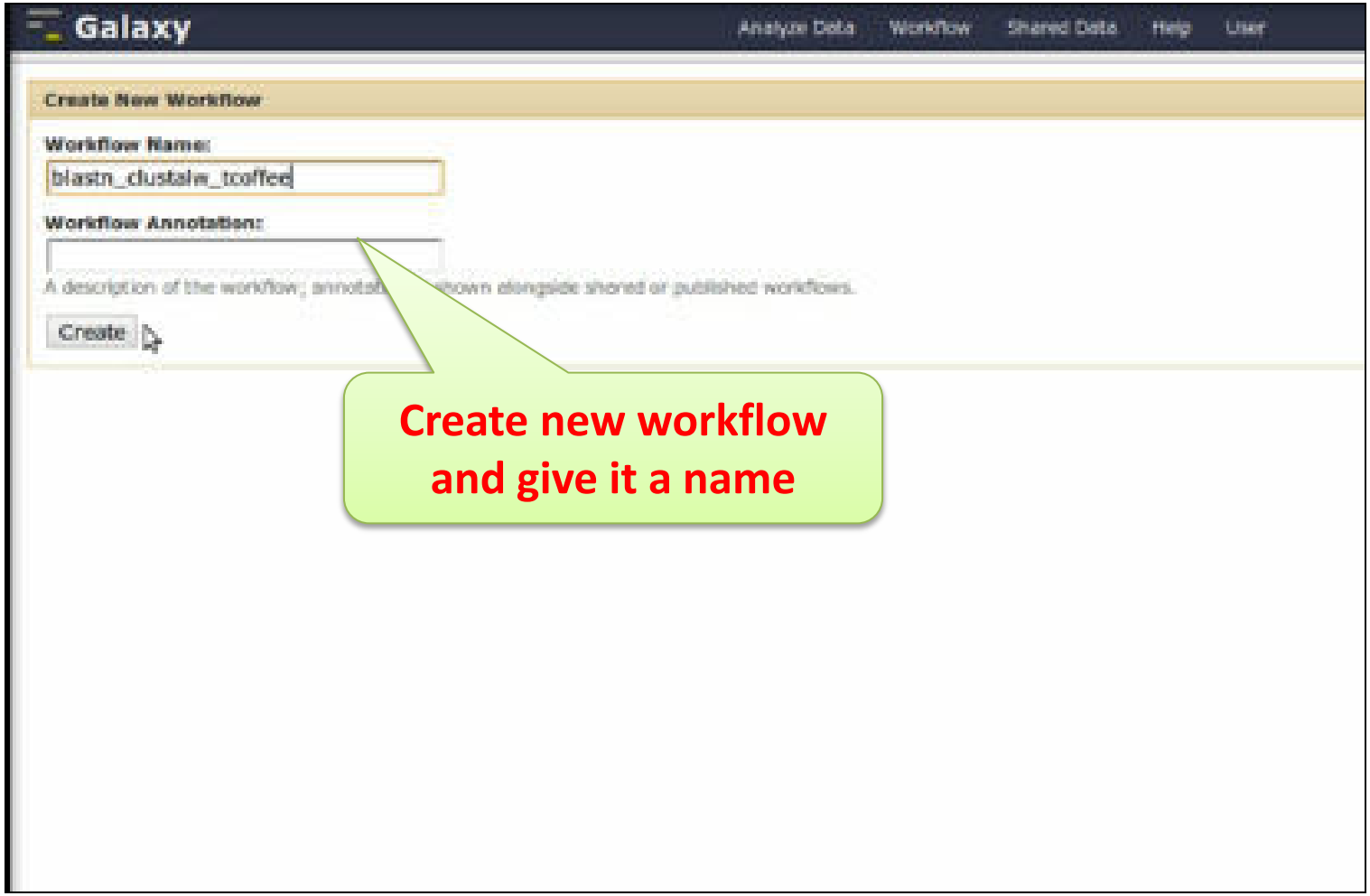
Most popular in Bioinformatics

http://en.wikipedia.org/wiki/Bioinformatics_workflow_management_systems

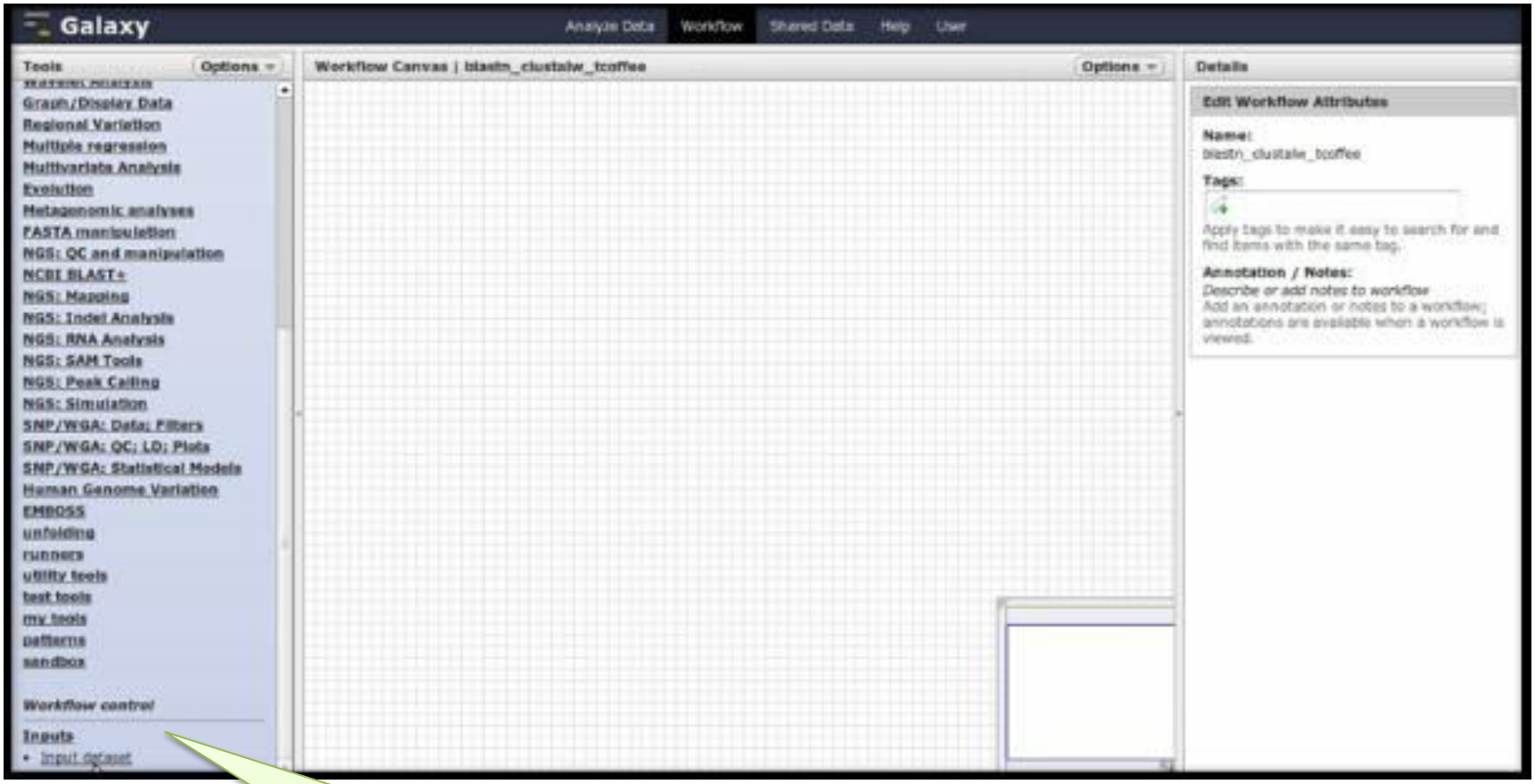
Composing a Workflow using Galaxy



Composing a Workflow using Galaxy

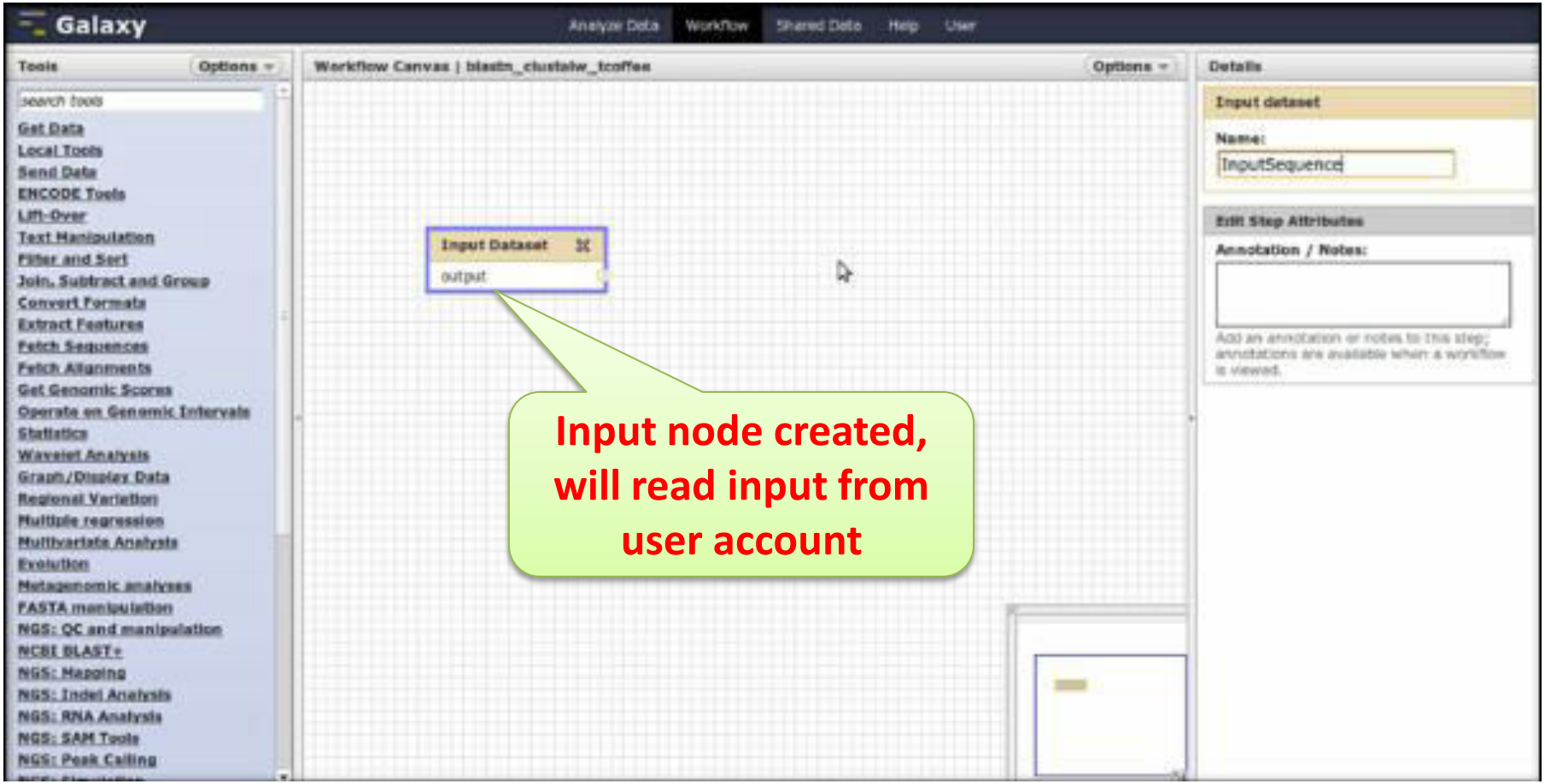


Composing a Workflow using Galaxy

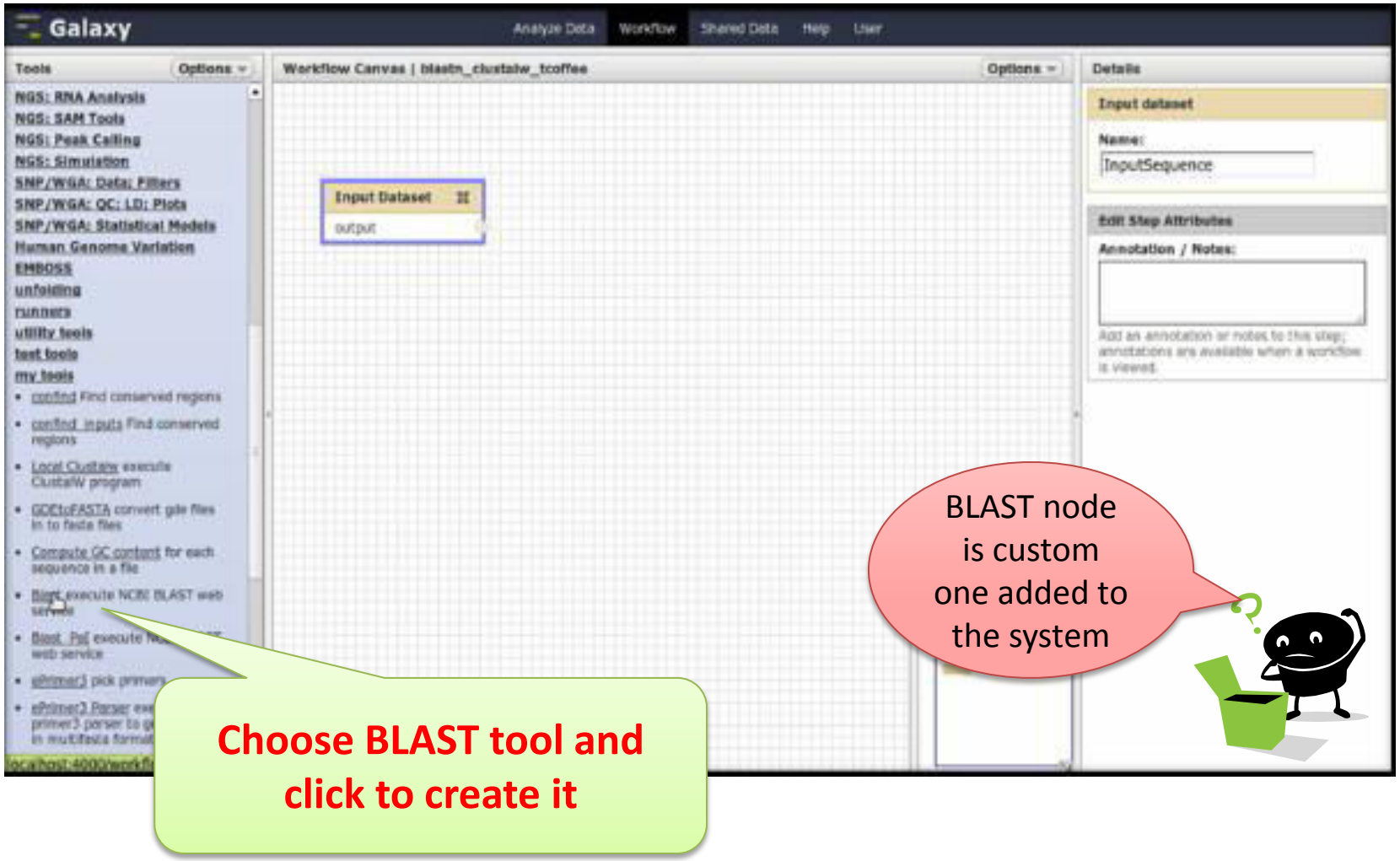


Click to create input node

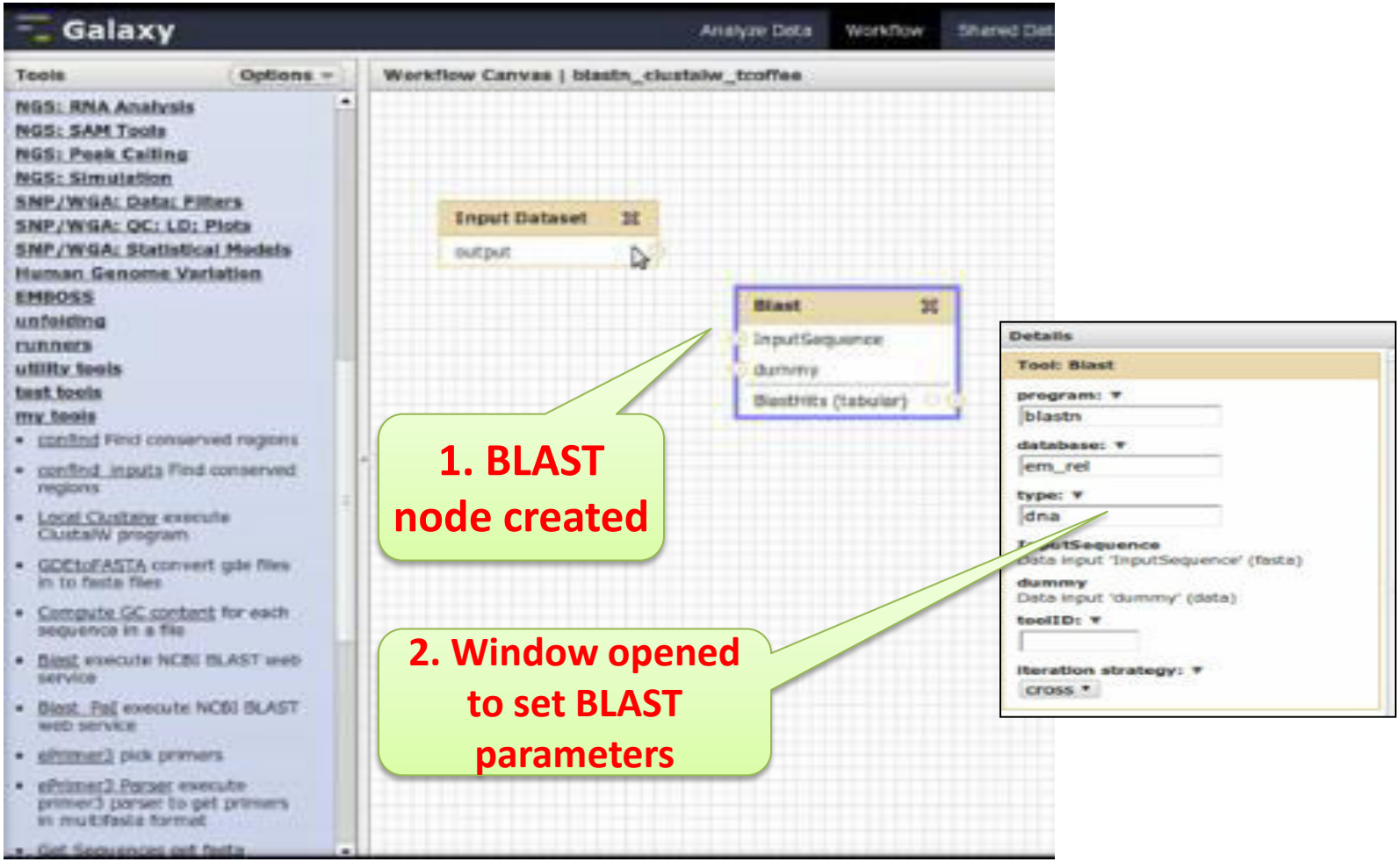
Composing a Workflow using Galaxy



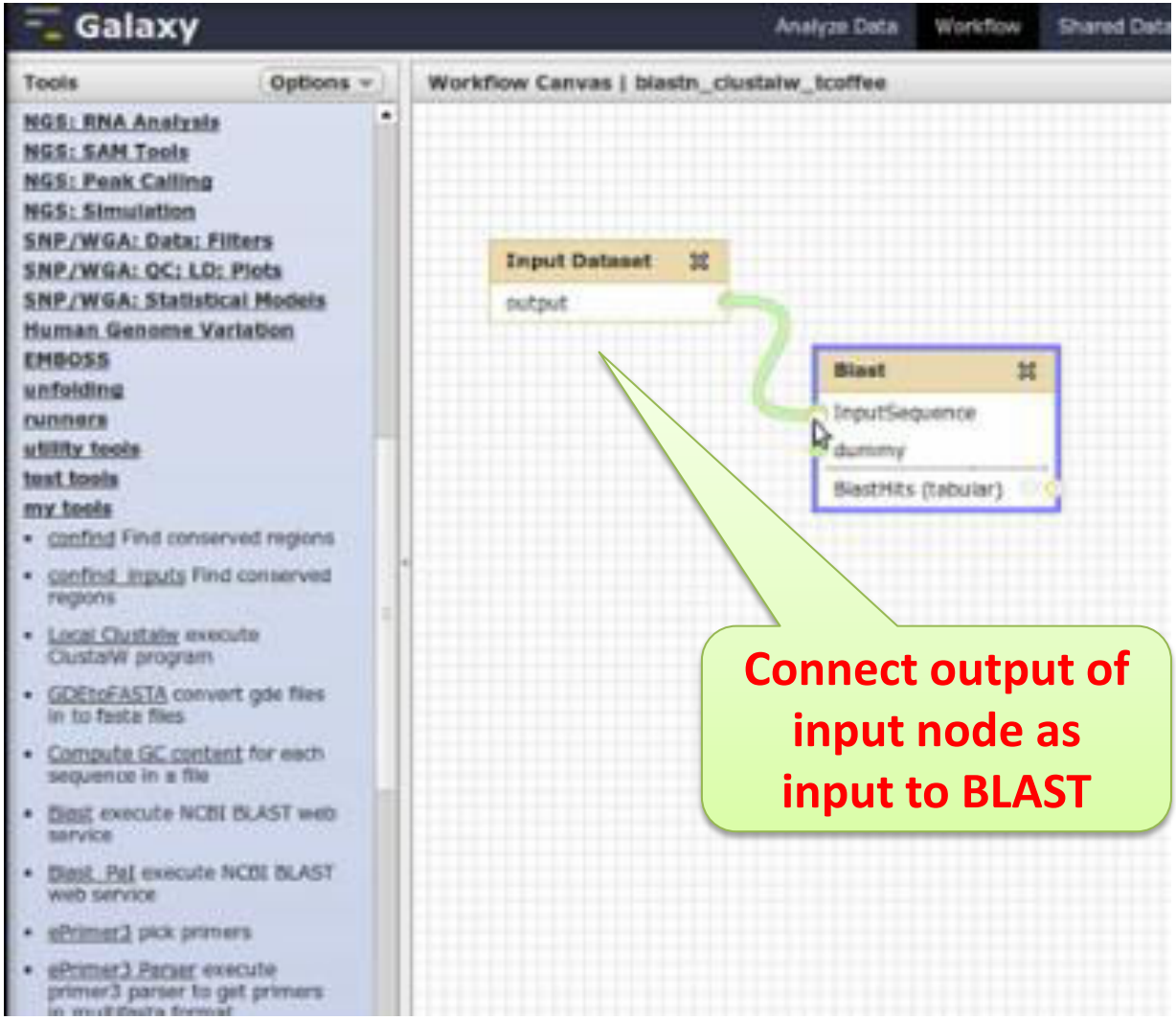
Composing a Workflow using Galaxy



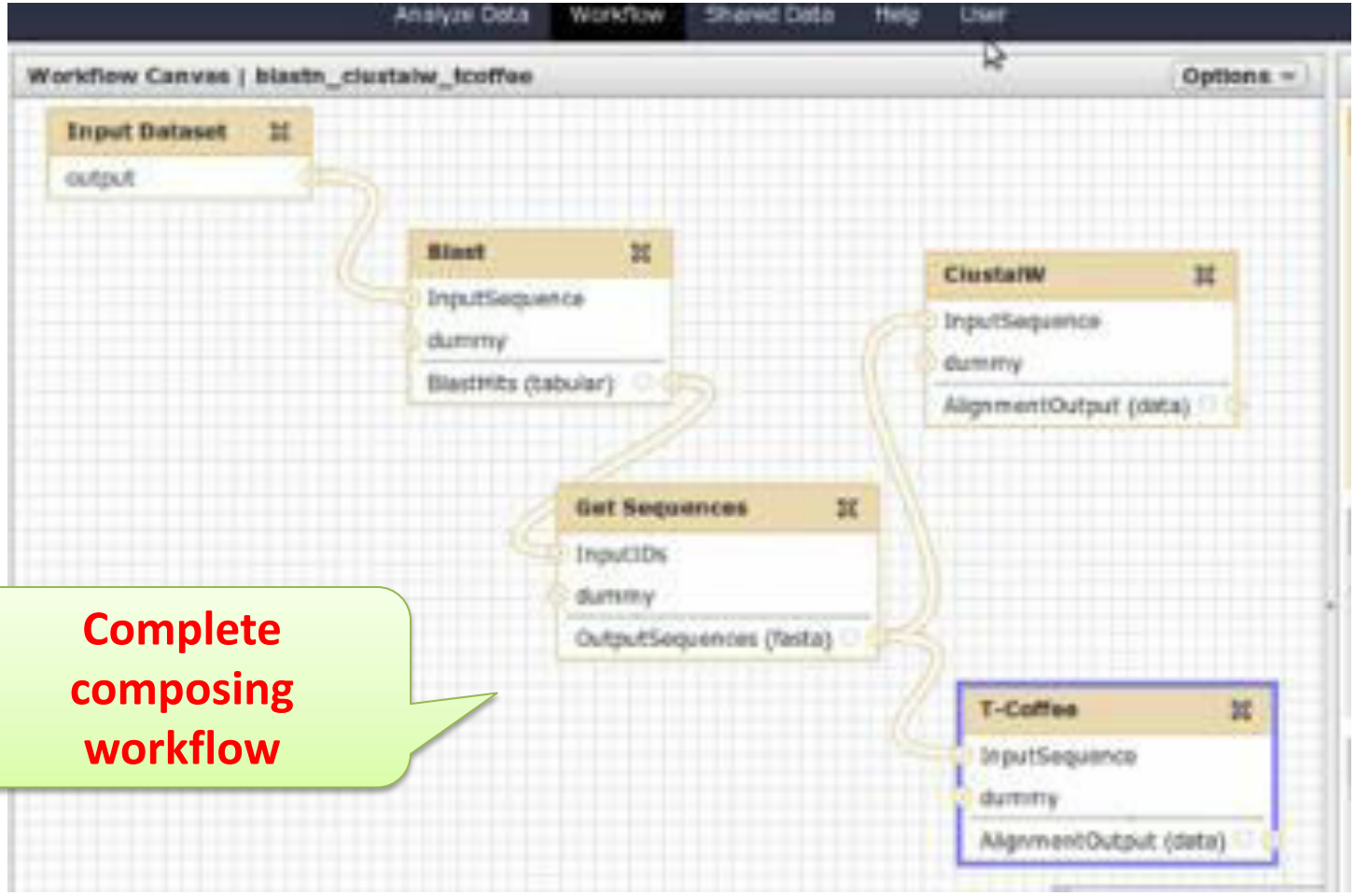
Composing a Workflow using Galaxy



Composing a Workflow using Galaxy

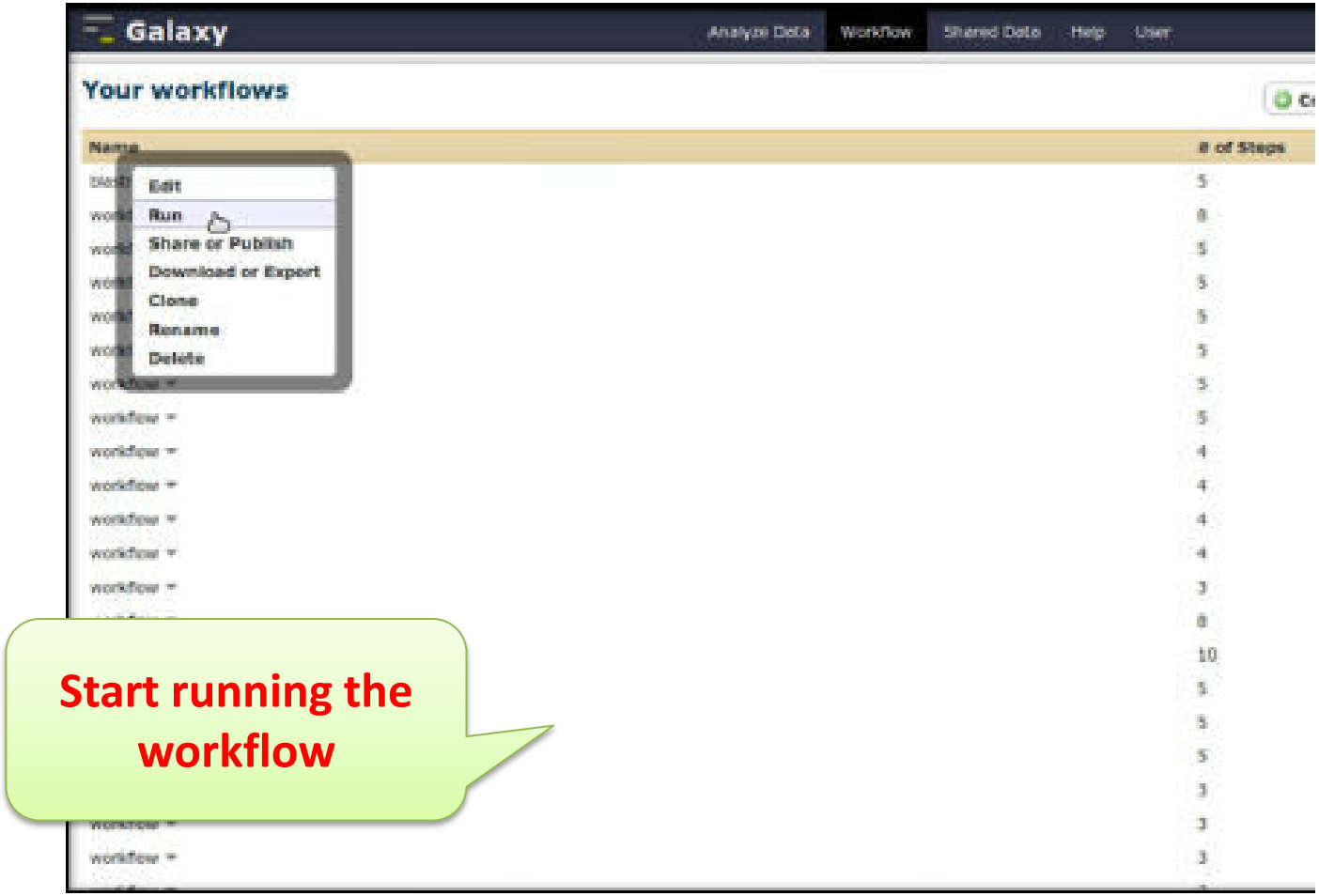


Composing a Workflow using Galaxy

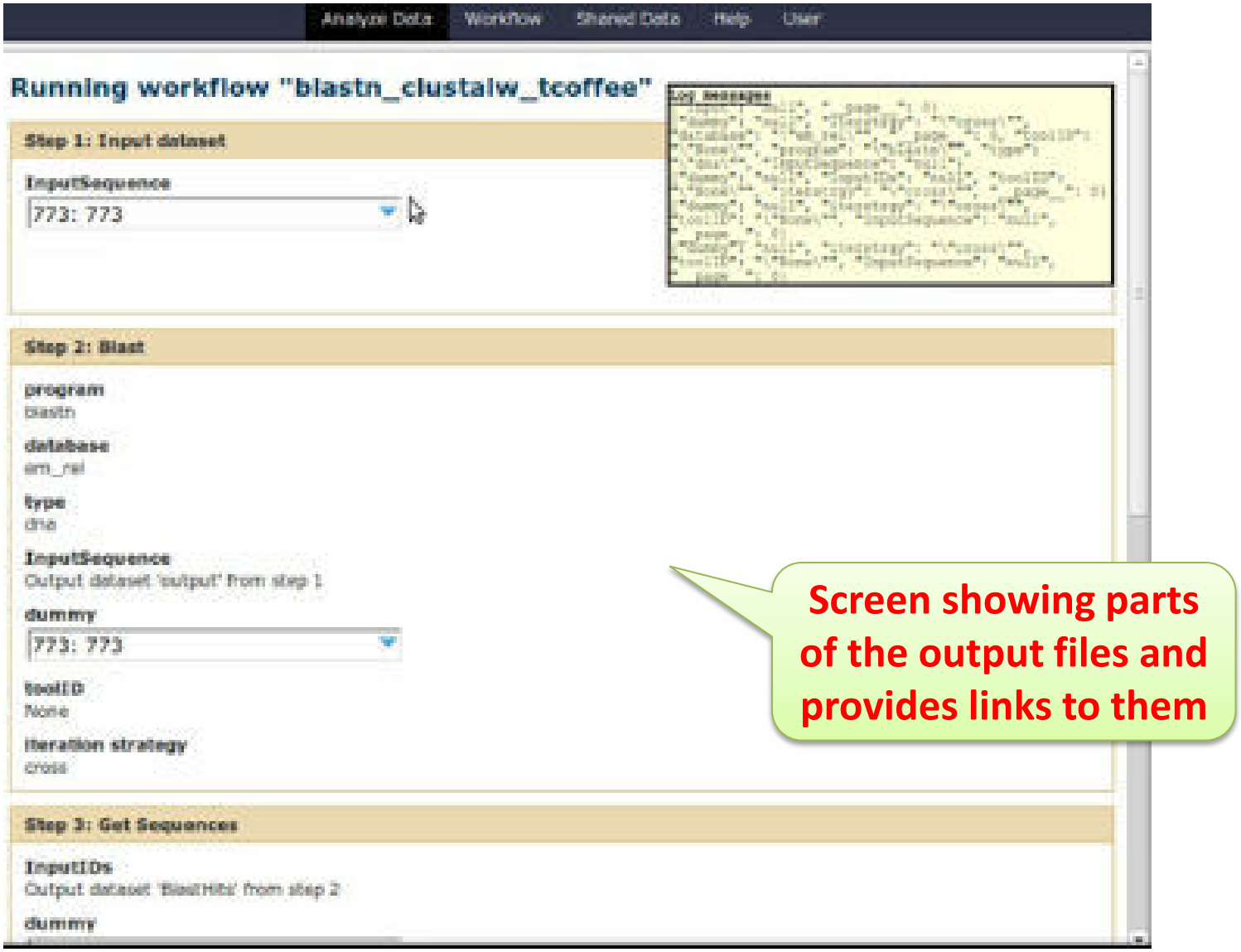


Complete composing workflow

Composing a Workflow using Galaxy



Composing a Workflow using Galaxy



Screen showing parts of the output files and provides links to them

Benefits of Workflow Systems

Accelerates computation

- Intuitive abstract means for describing computational experiments
- Requires no programming expertise
- Easy to modify
- Hide execution details: invocation and scheduling
- Direct use of parallel architectures

Benefits of Workflow Systems

Formalize computation

- Come with library of tools or access directories
  → Tool accessibility

- Store experiment details (used tools, their parameters, and used data)
  → Reproducibility

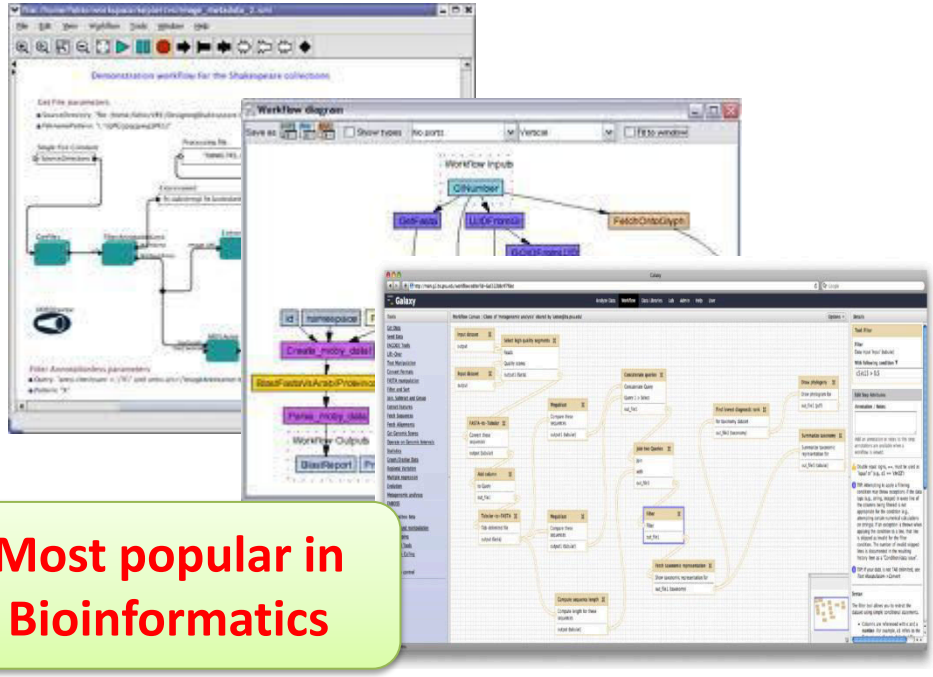
- Share workflows with analysis history including intermediate results
  → Transparency

Mesirov. Science, 2010
B. Giardine, et al. Genome Research, 2005.

Implementing Scientific Workflows

Methods 2: Use of Workflow Systems

- Kepler
- Triana
- WildFire
- GenePattern
- Pegsus
- Taverna
- Galaxy
- and many more

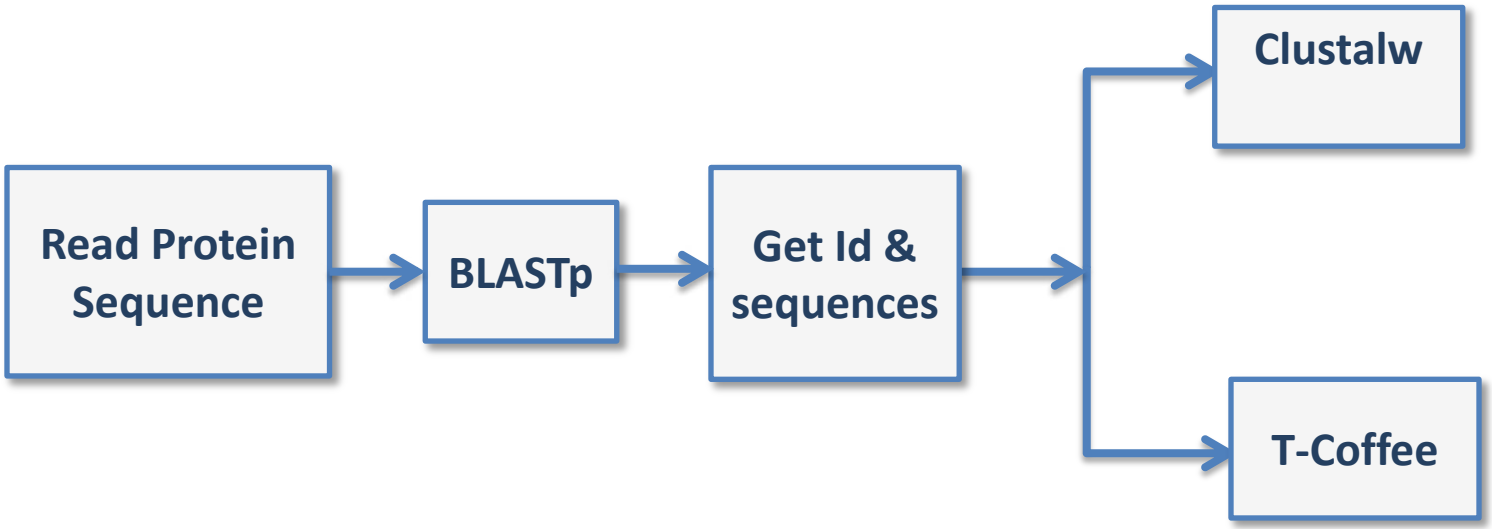


Most popular in Bioinformatics

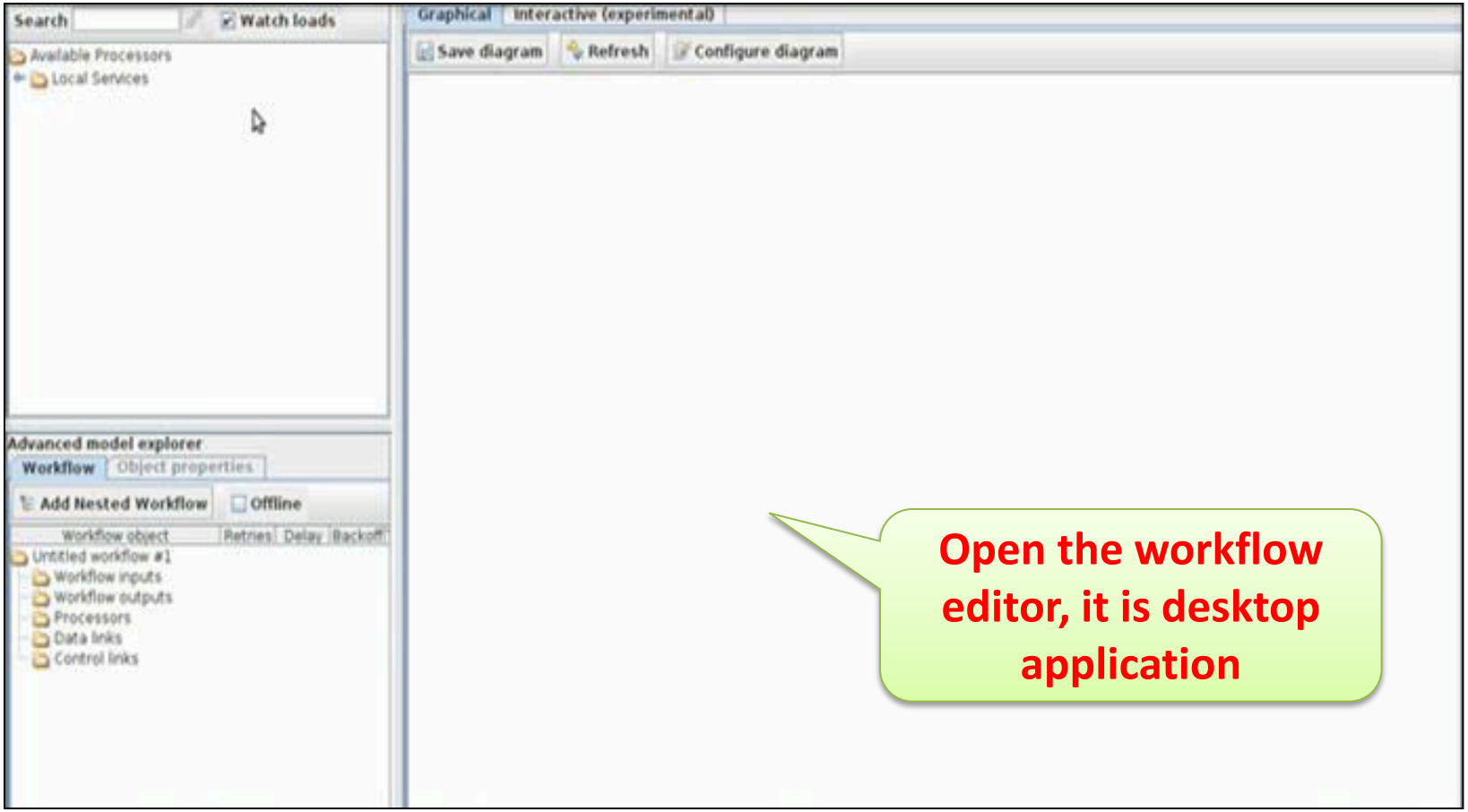
http://en.wikipedia.org/wiki/Bioinformatics_workflow_management_systems

Composing a Workflow using Taverna

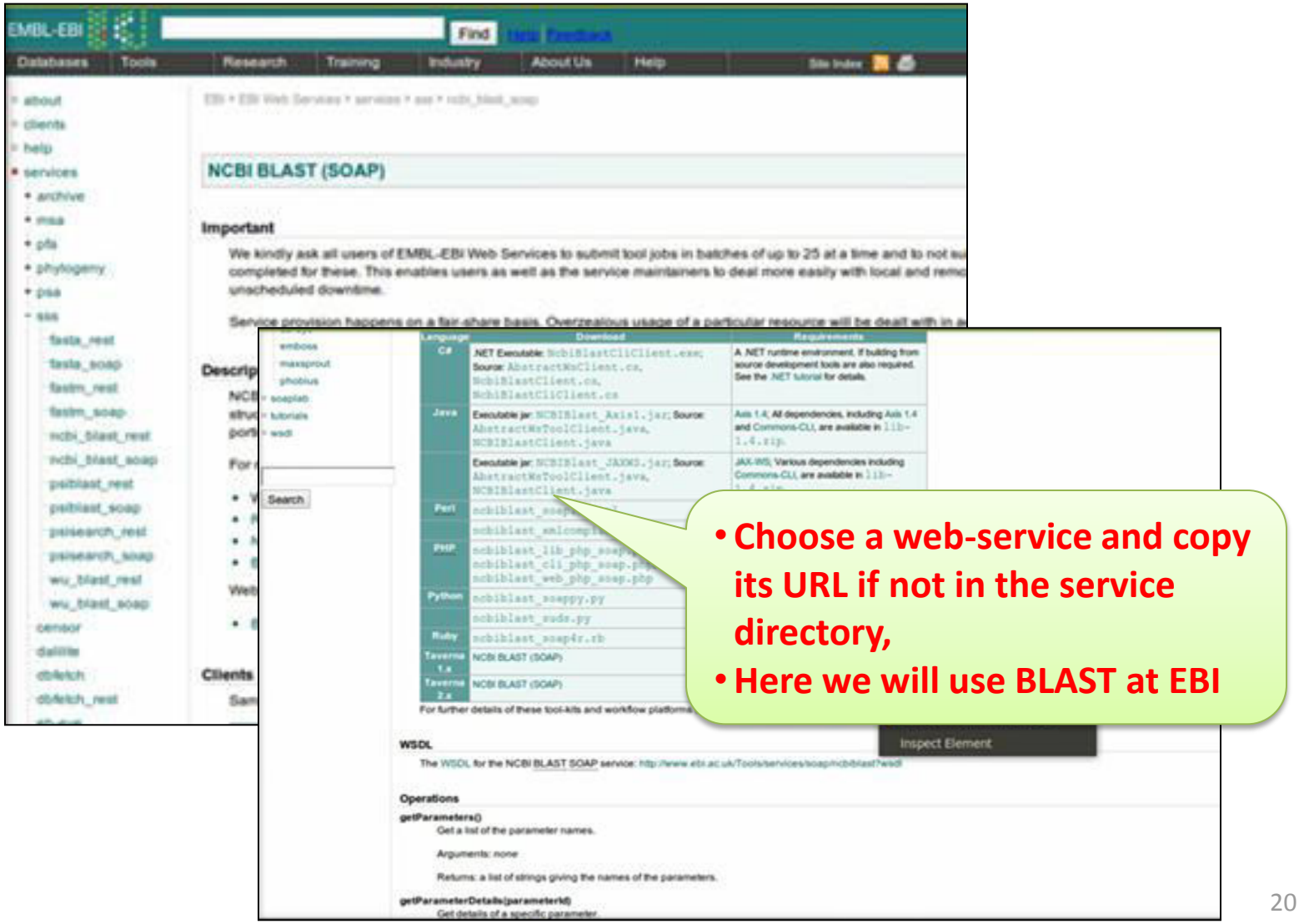
Example: Homology Workflow



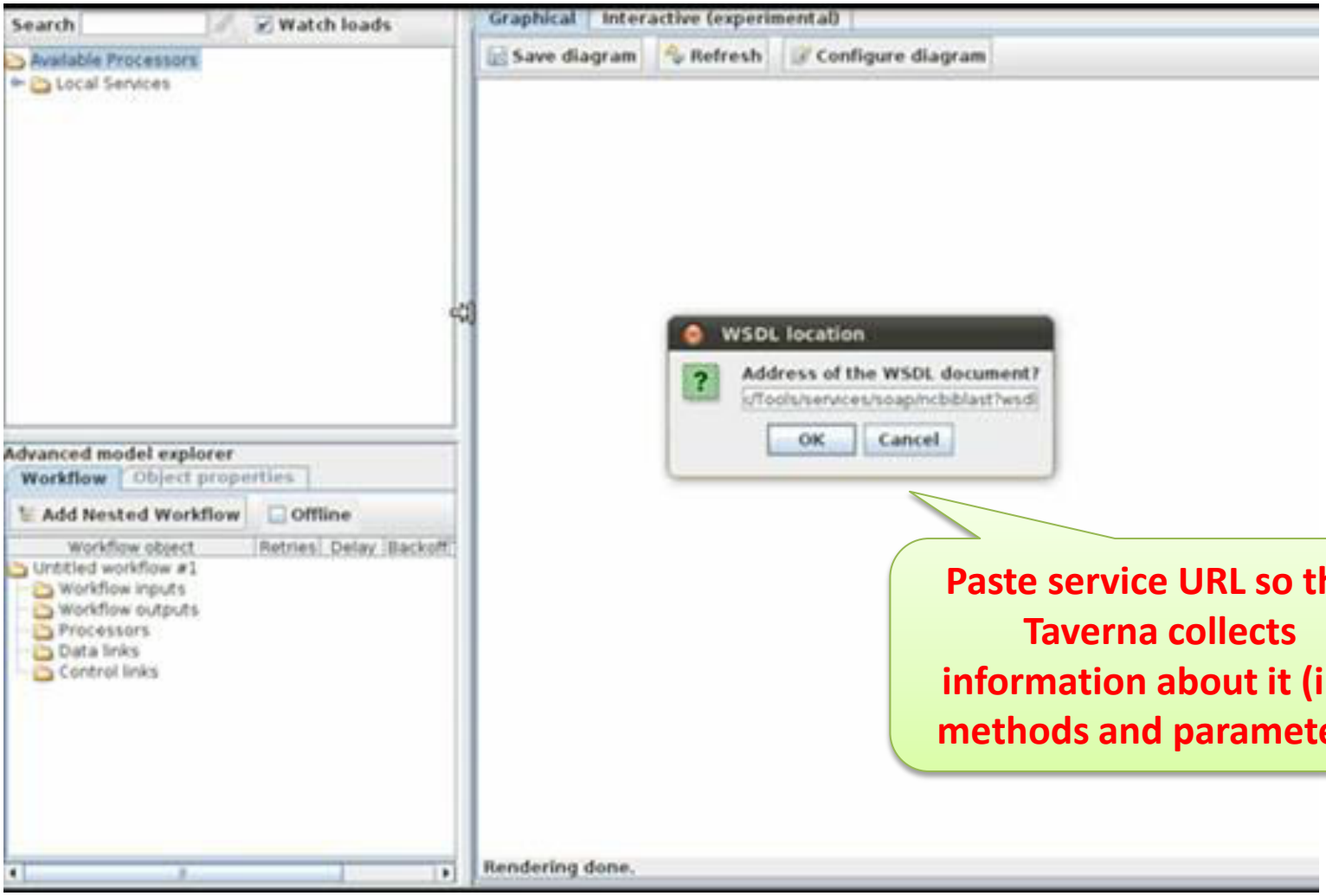
Composing a Workflow using Taverna



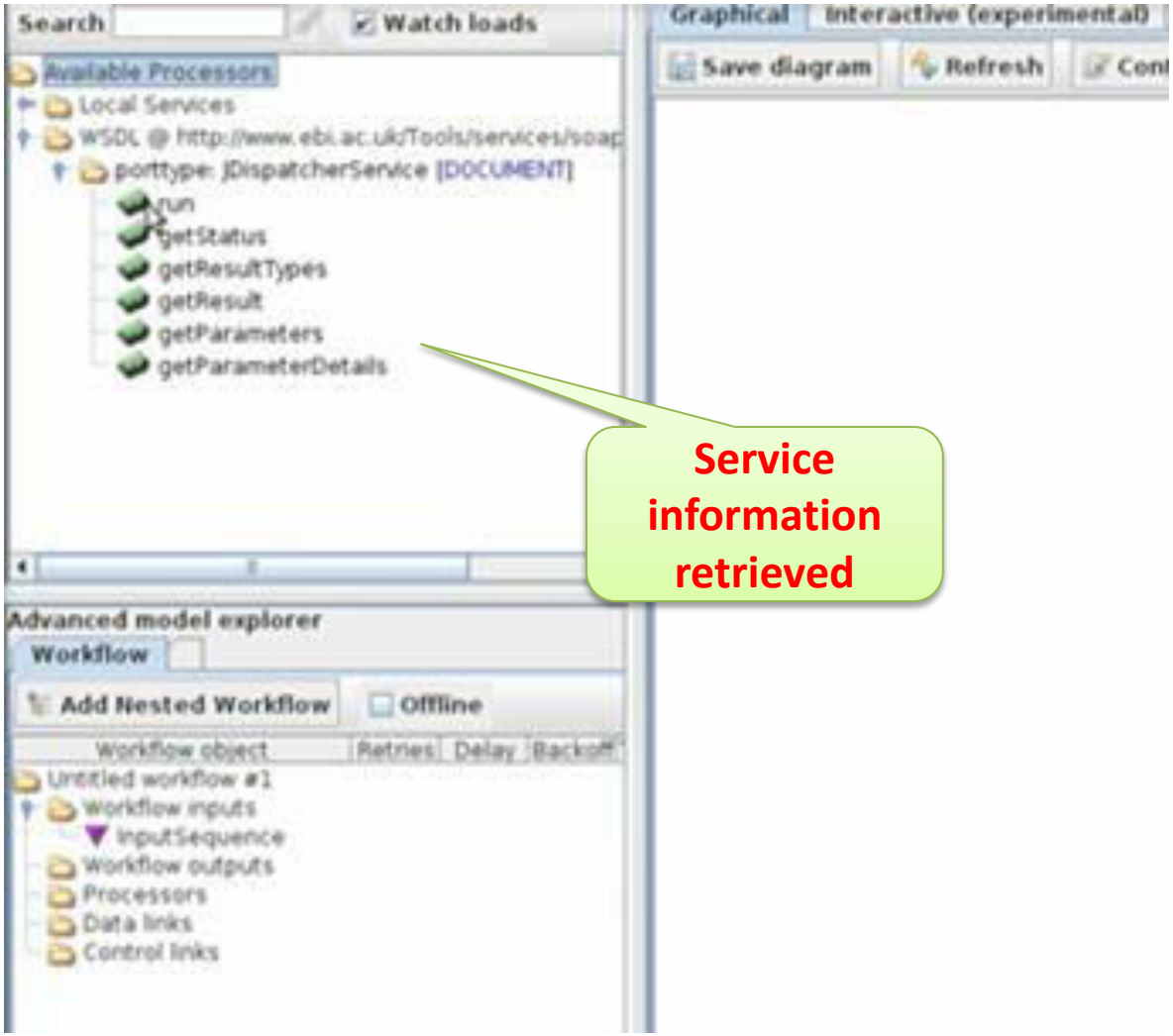
Composing a Workflow using Taverna



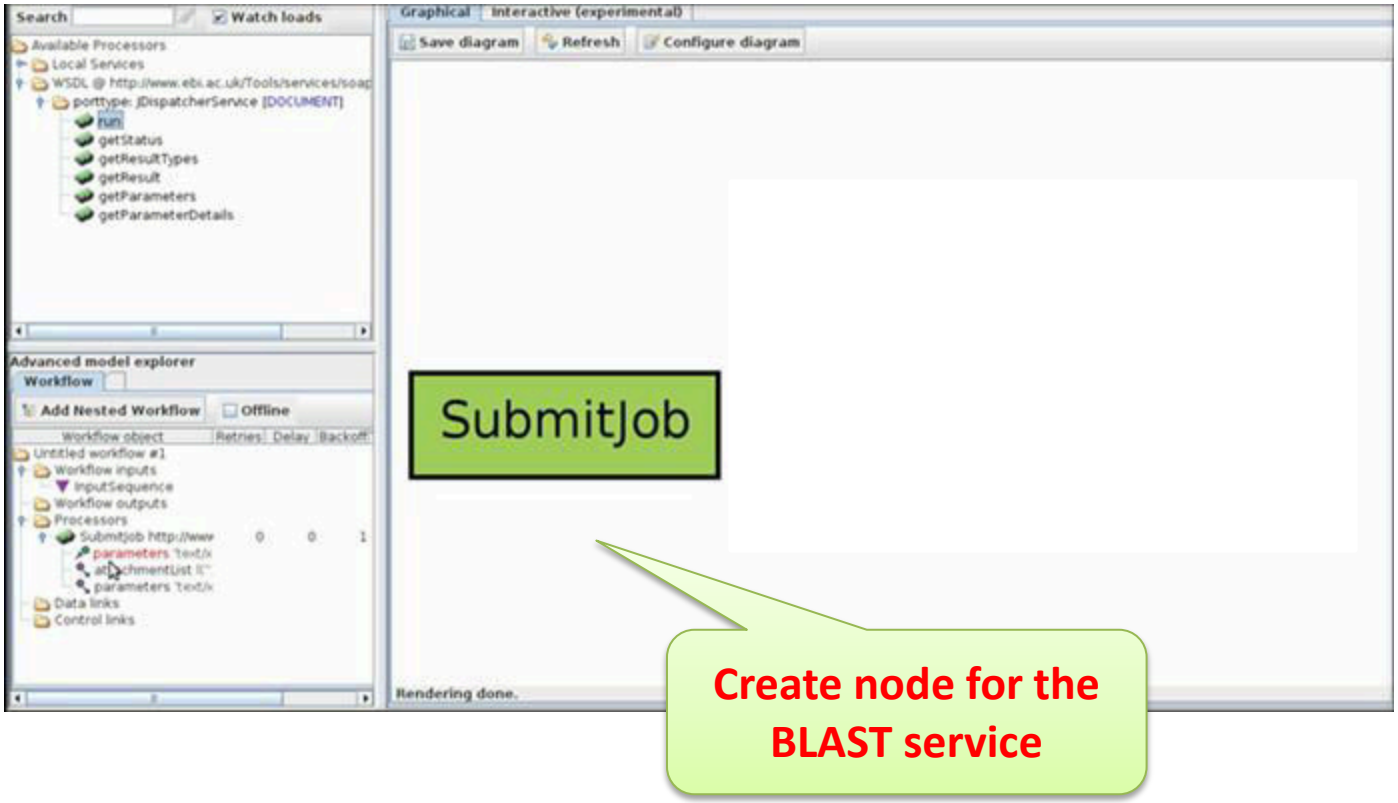
Composing a Workflow using Taverna



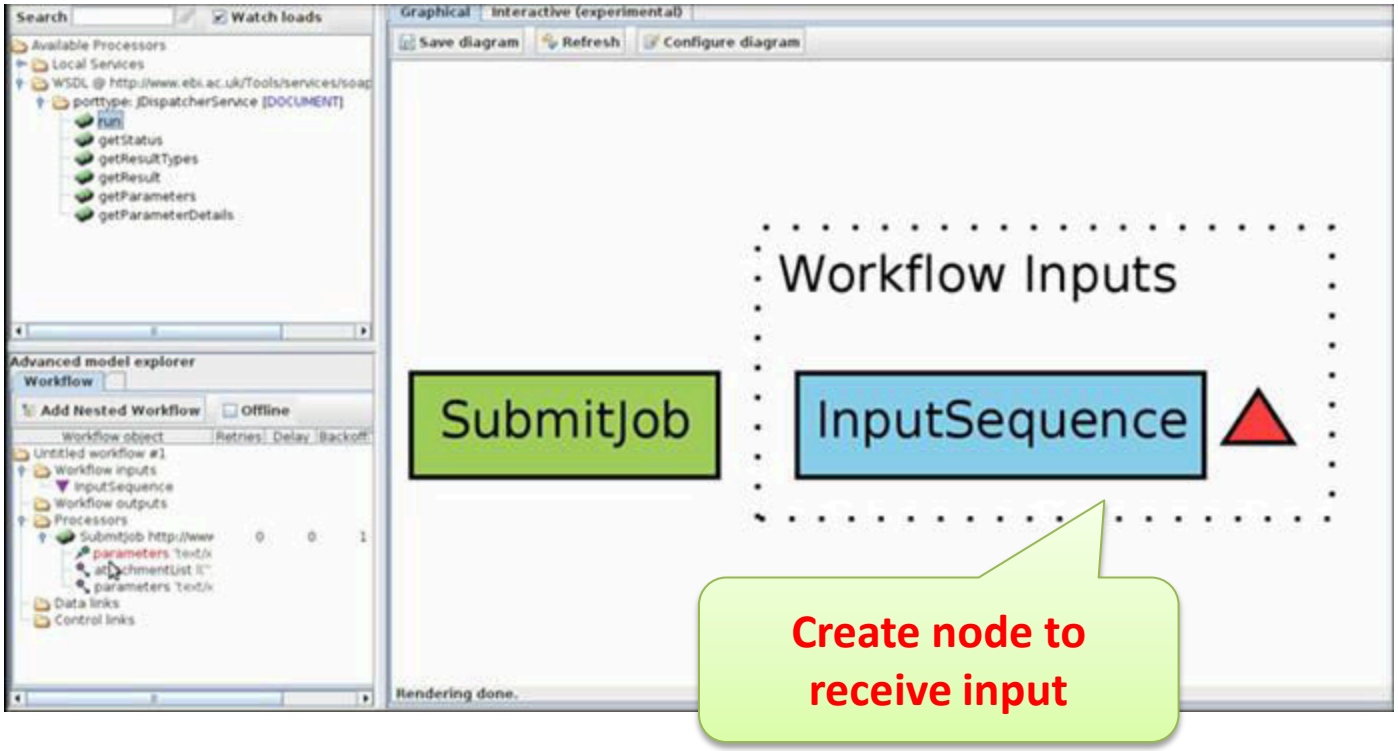
Composing a Workflow using Taverna



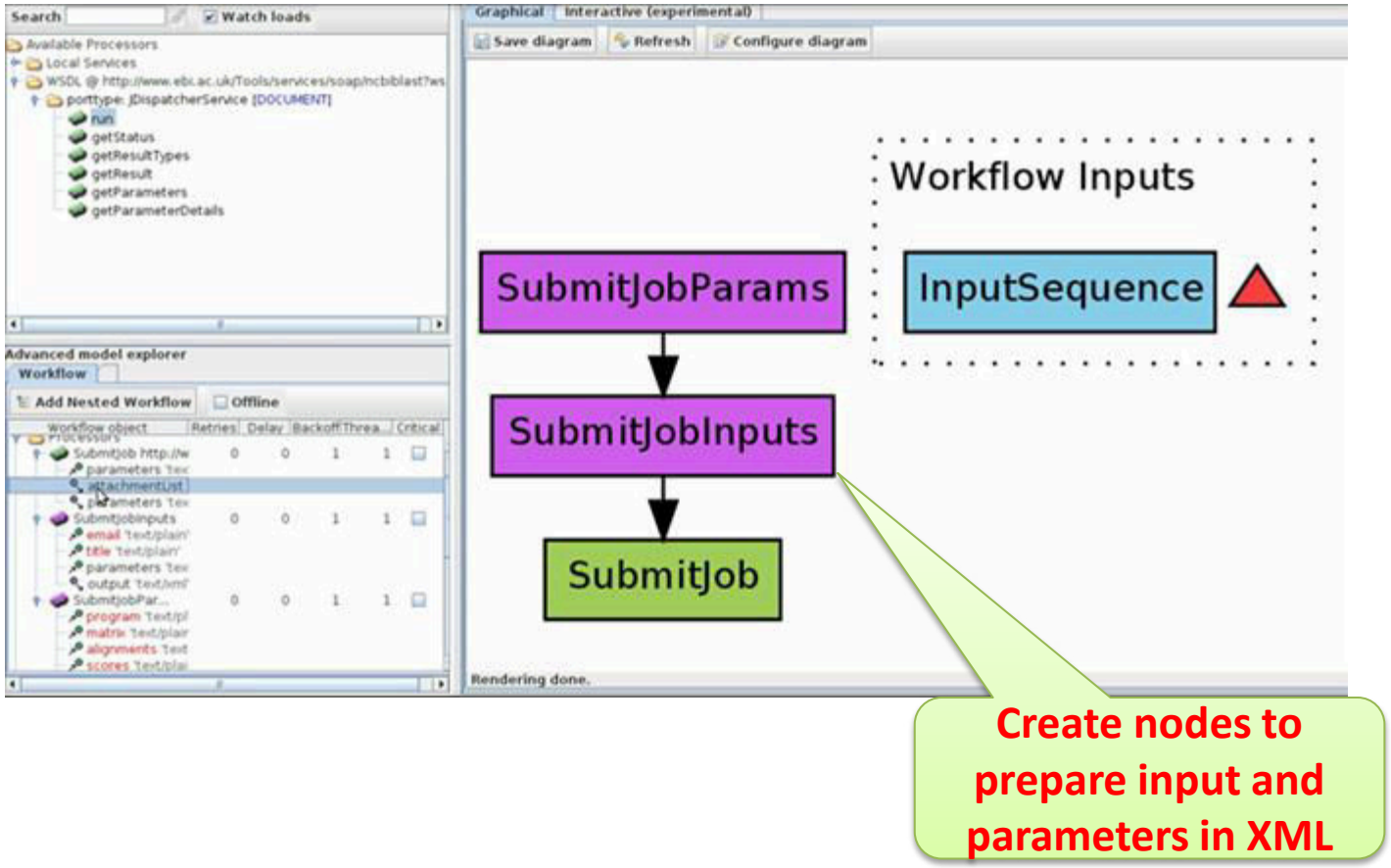
Composing a Workflow using Taverna



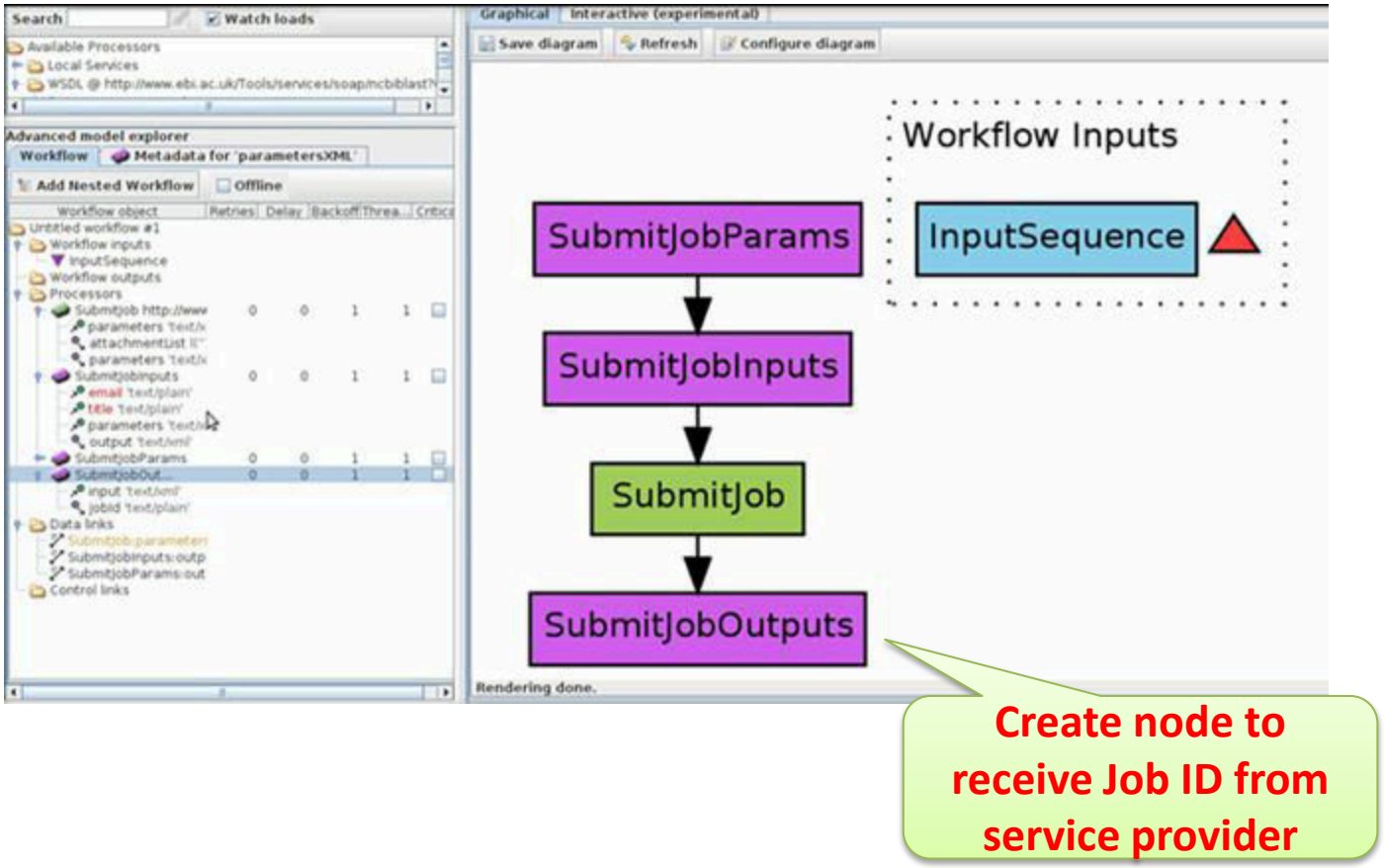
Composing a Workflow using Taverna



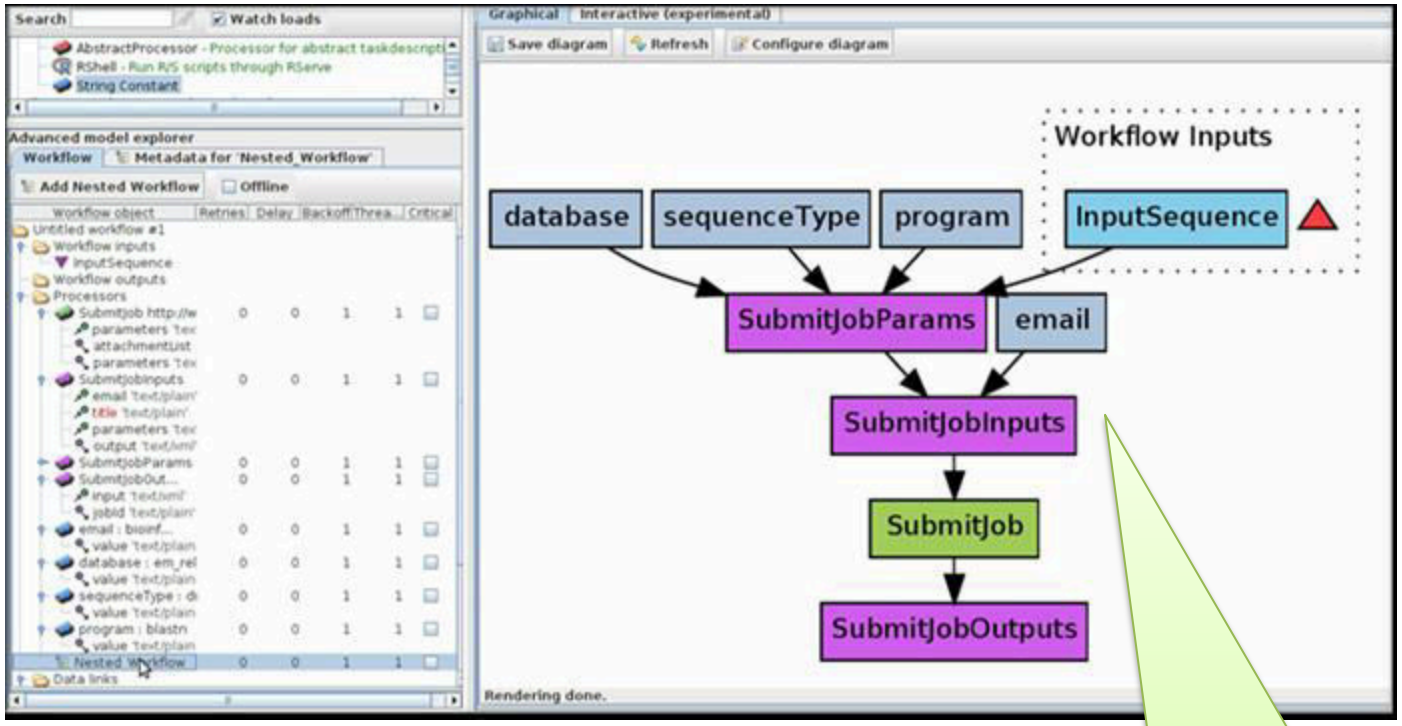
Composing a Workflow using Taverna



Composing a Workflow using Taverna

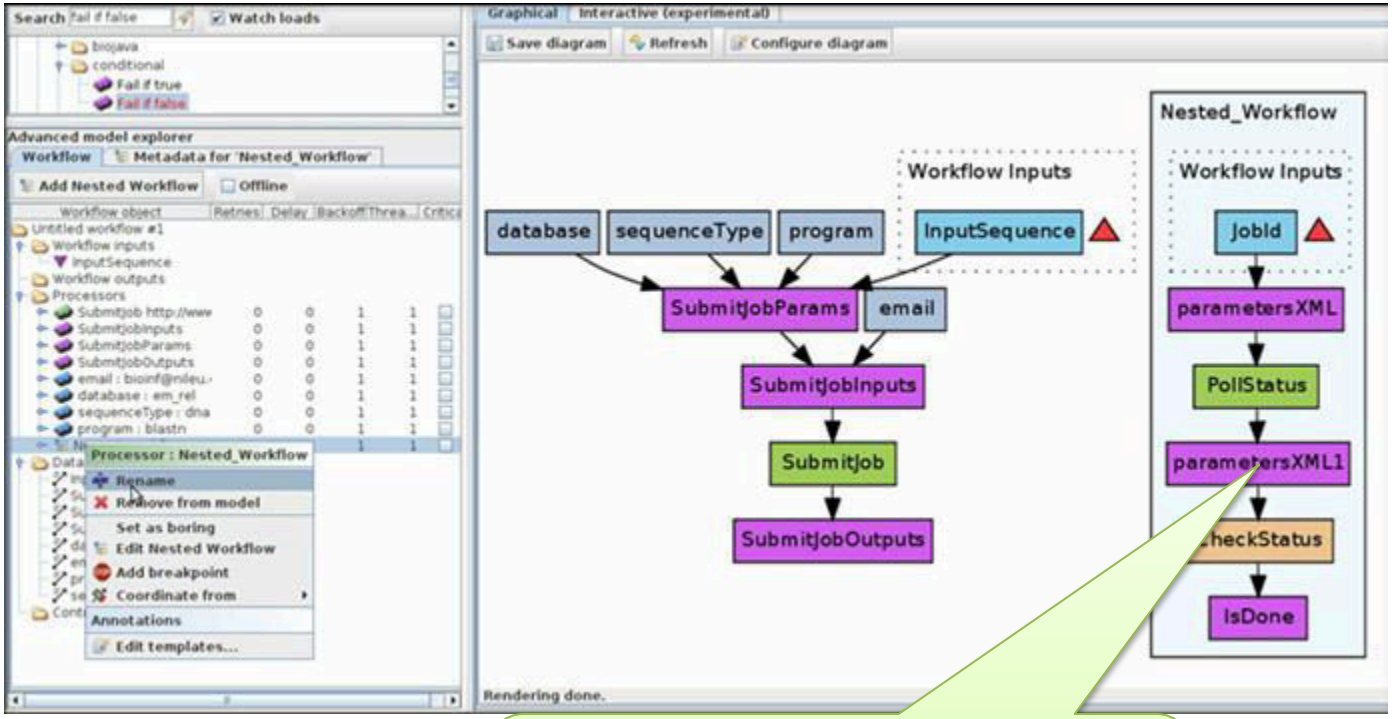


Composing a Workflow using Taverna



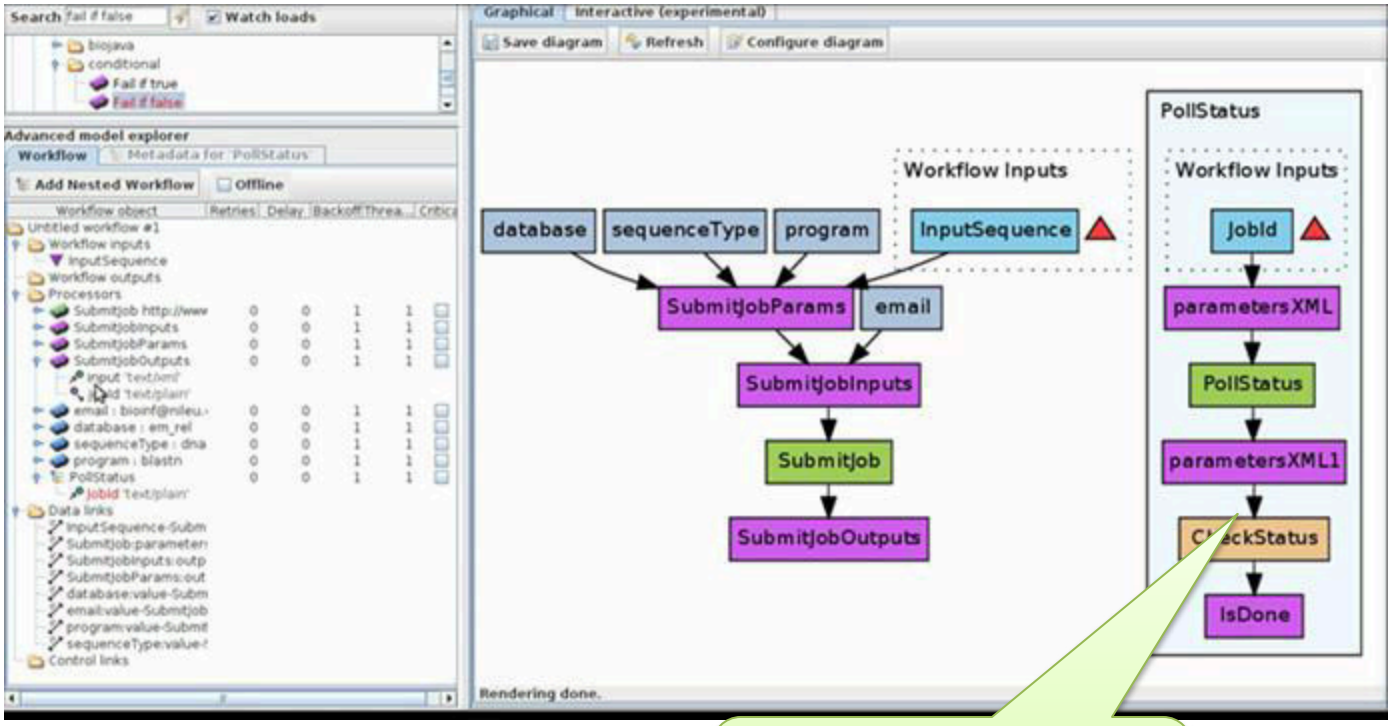
Input data and parameters are merged in XML file

Composing a Workflow using Taverna



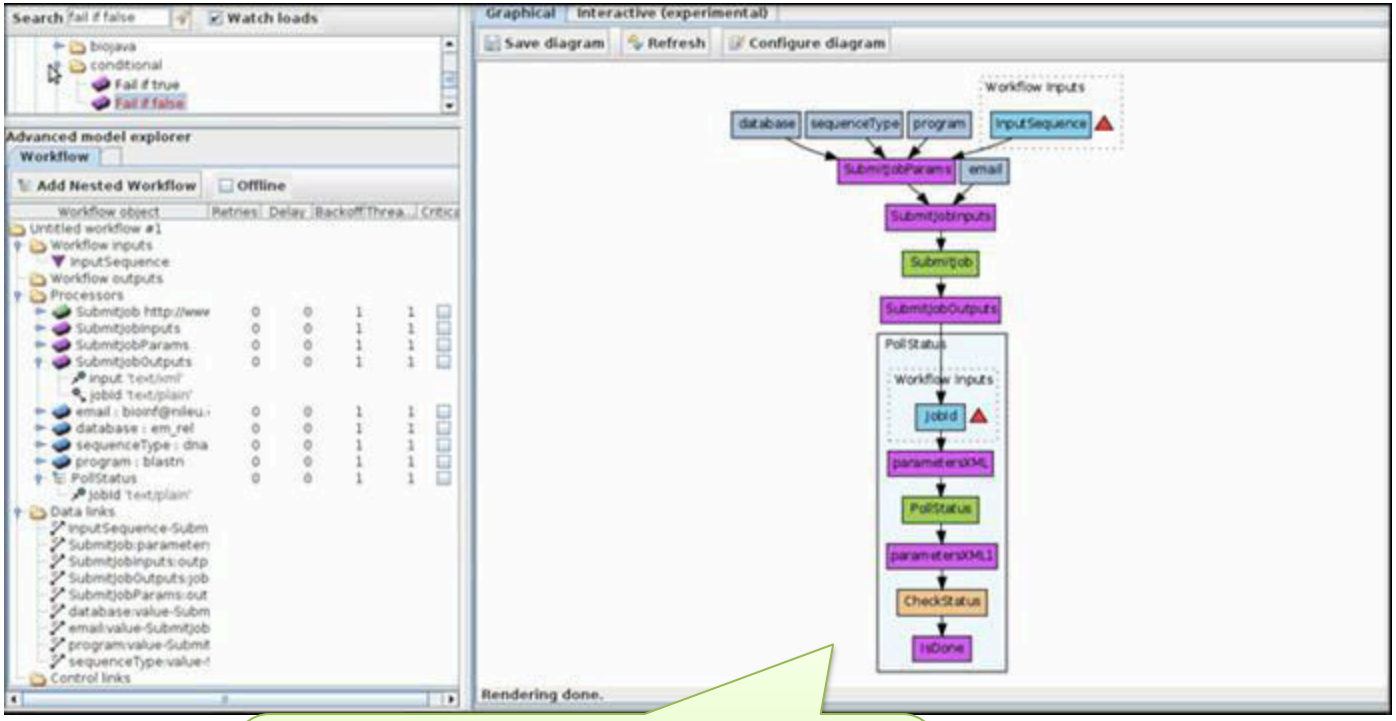
Create sub-workflow to check status of the asynchronous service

Composing a Workflow using Taverna



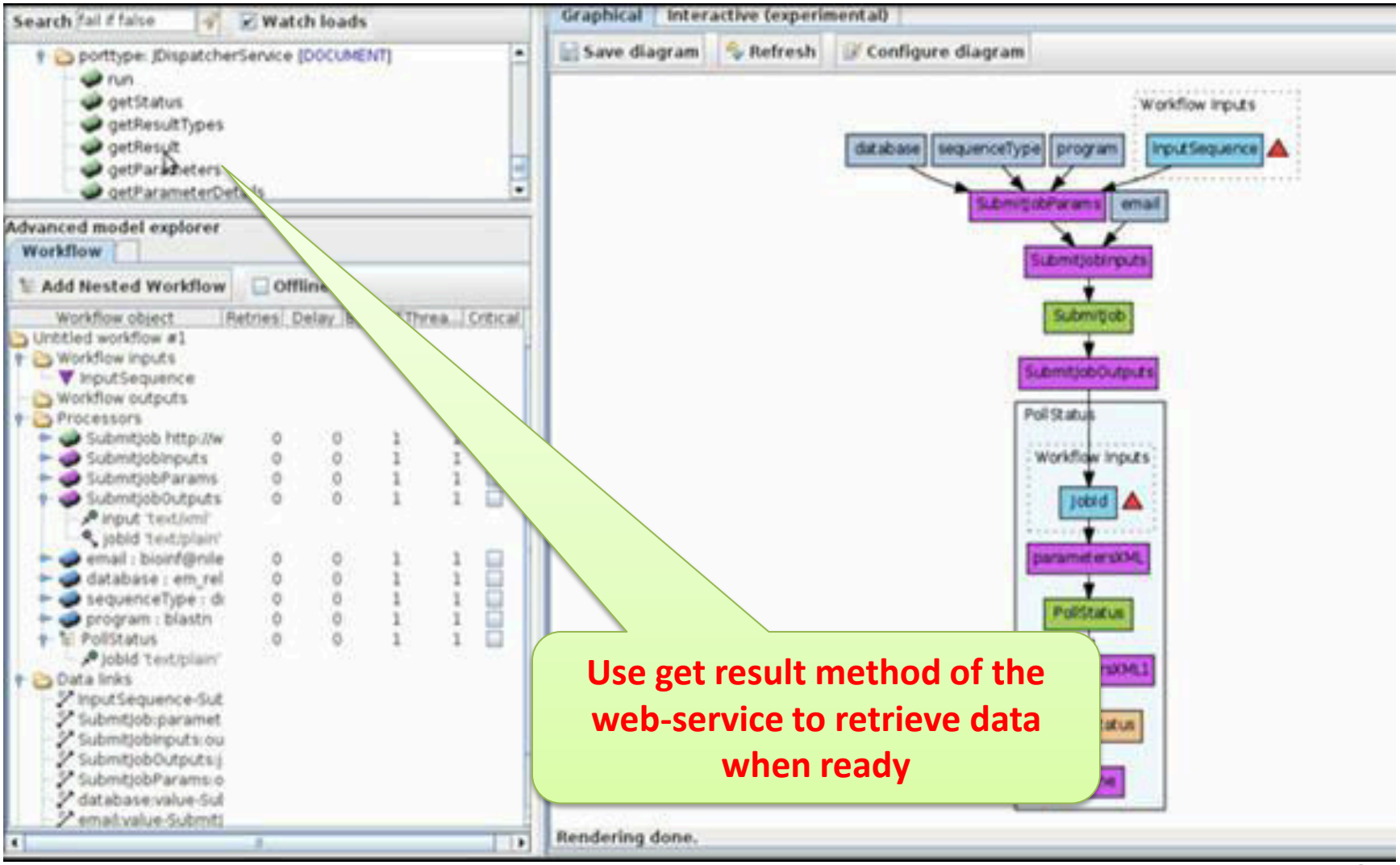
Rename it Poll status

Composing a Workflow using Taverna

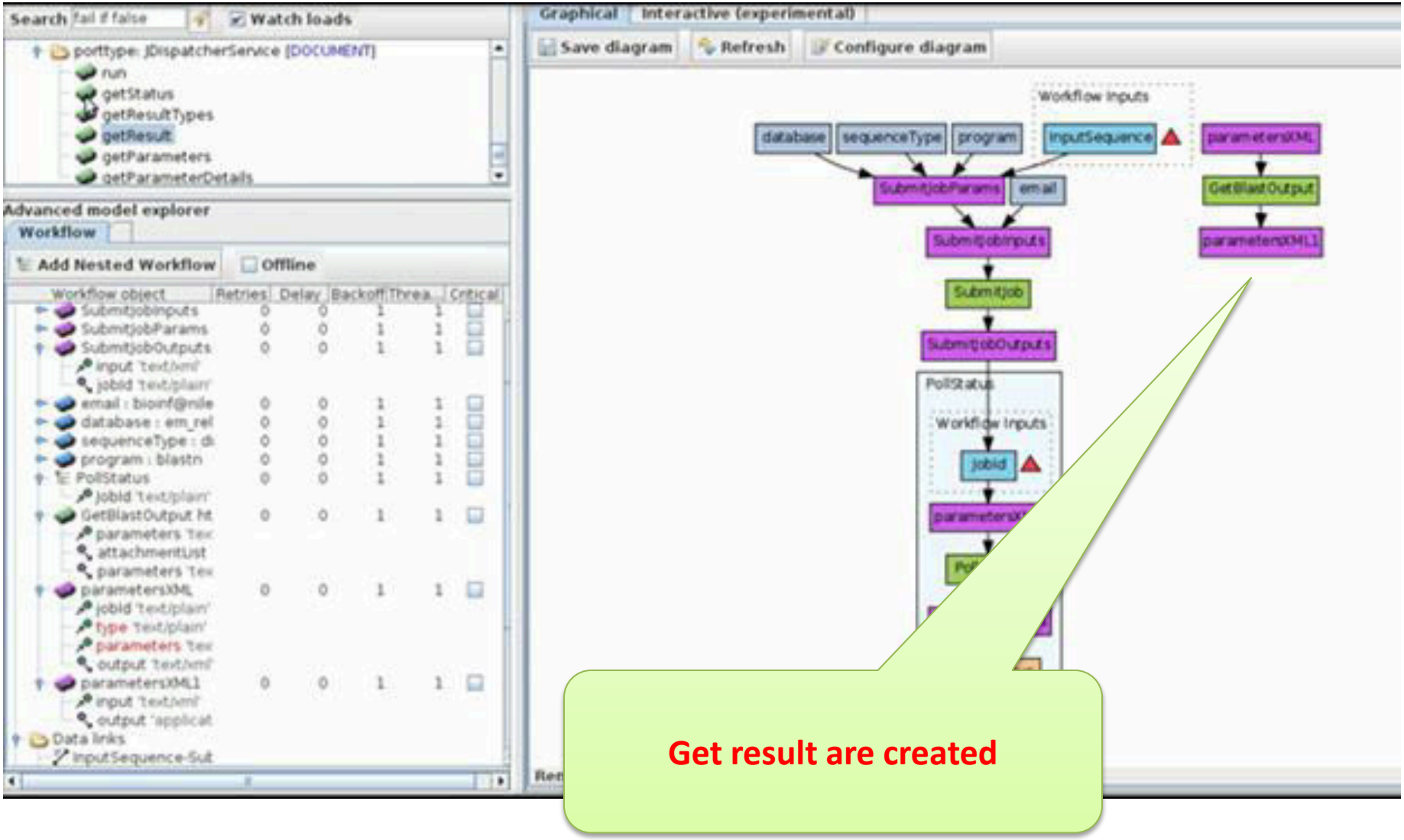


Put in place after job submission

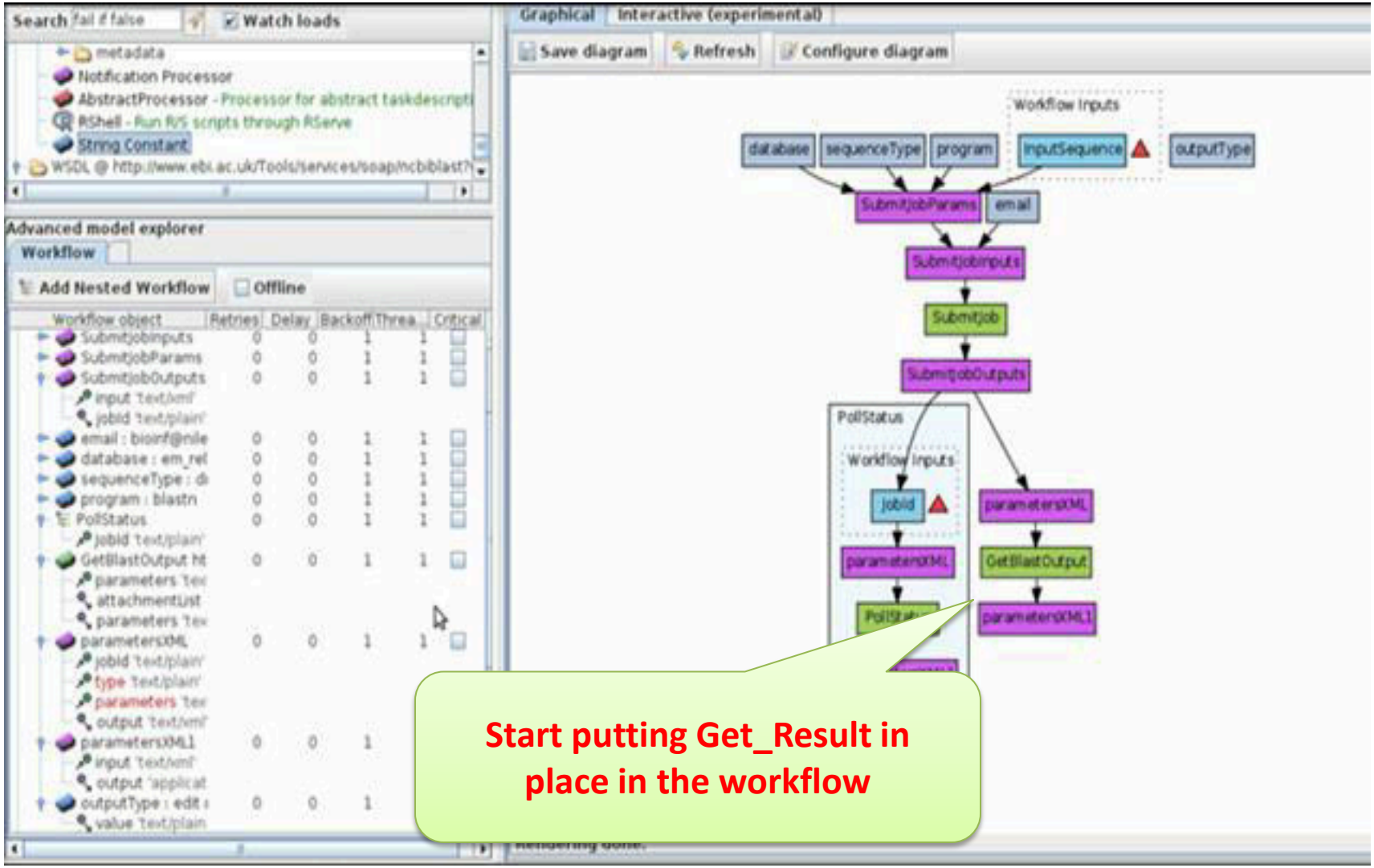
Composing a Workflow using Taverna



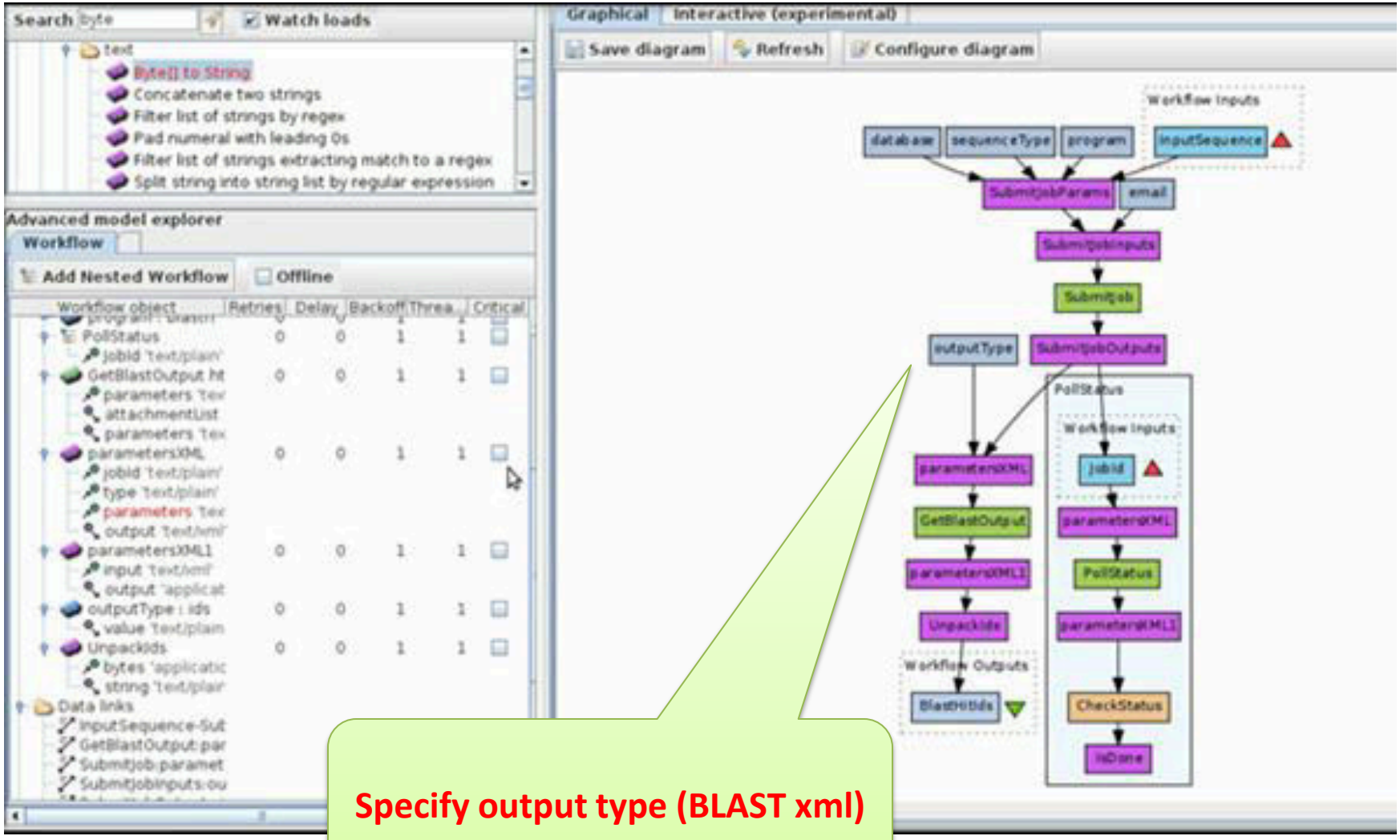
Composing a Workflow using Taverna



Composing a Workflow using Taverna

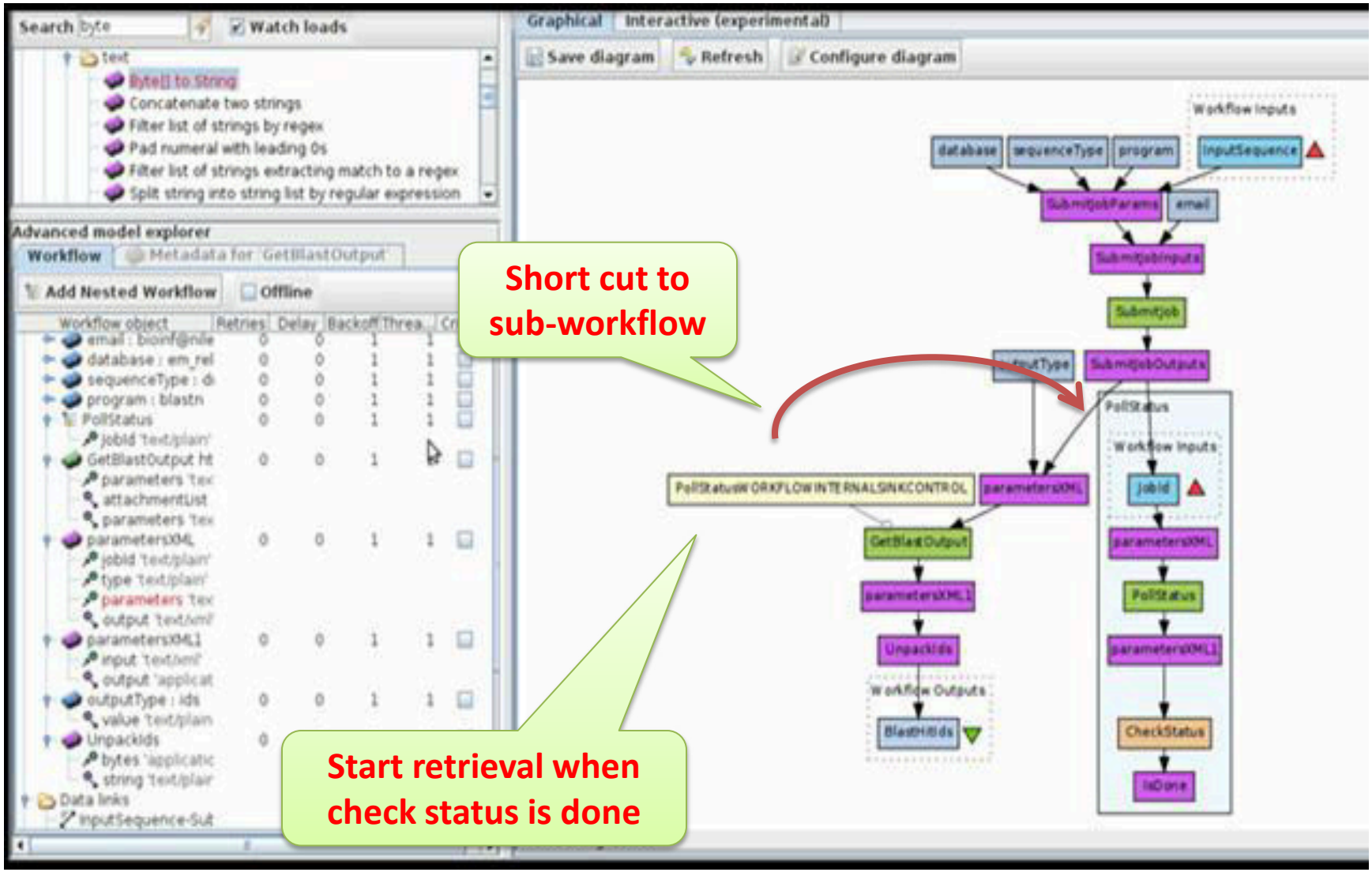


Composing a Workflow using Taverna

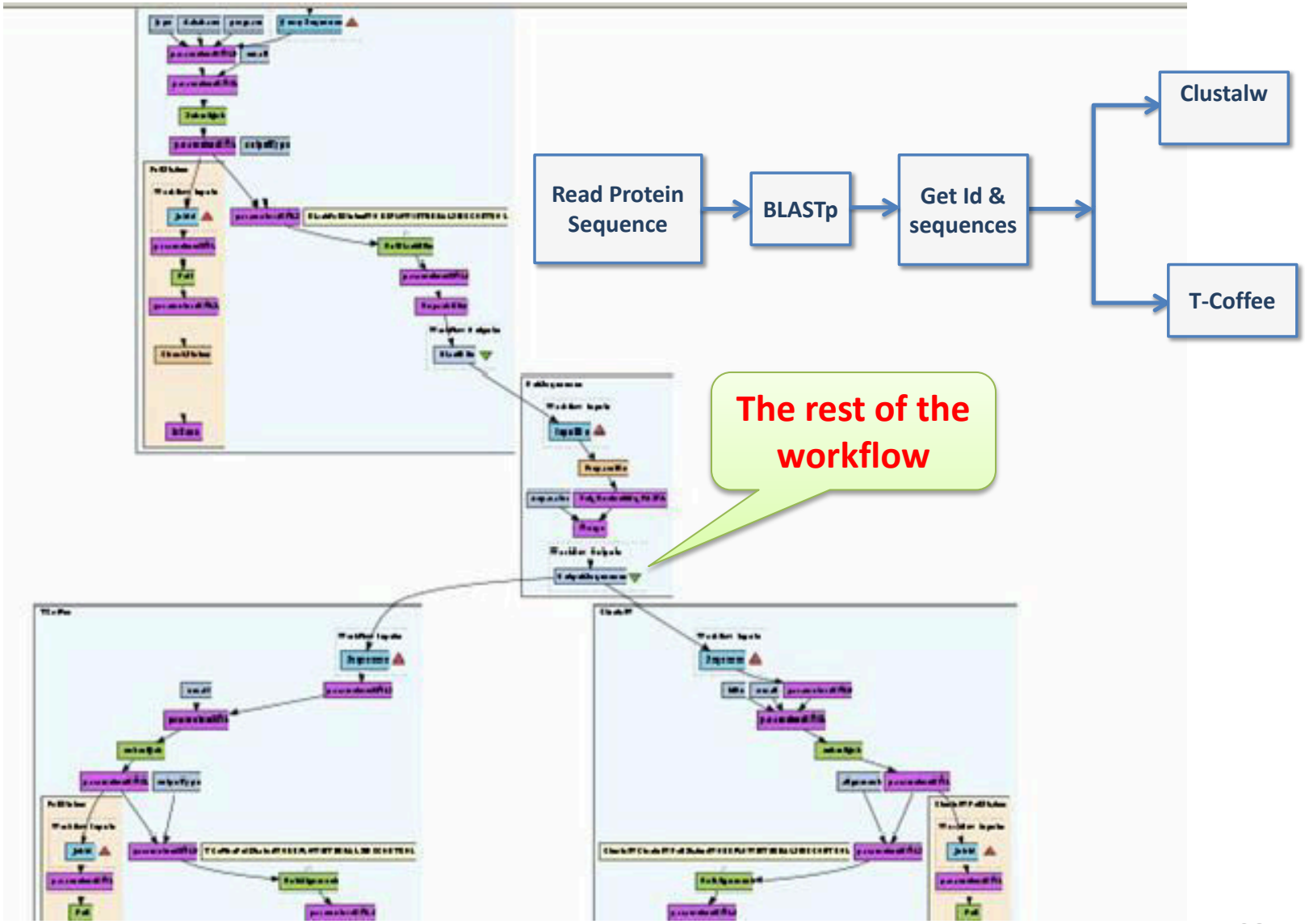


Specify output type (BLAST xml)

Composing a Workflow using Taverna



Composing a Workflow using Taverna



Taverna vs. Galaxy

	Taverna	Galaxy
Purpose	General	Bioinformatics
Interface	Desktop	Web-interface
Usability	More difficult	Easy
Engine	Control flow	Data flow
Control constructs	Yes	No
Jobs	Web-service	Local invocation
Programs	Service directory	Library of tools
Use of Local HPC	Threads only	Threads/Cluster

Taverna

D. Hull, et al. Nucleic Acids Research, 2006.

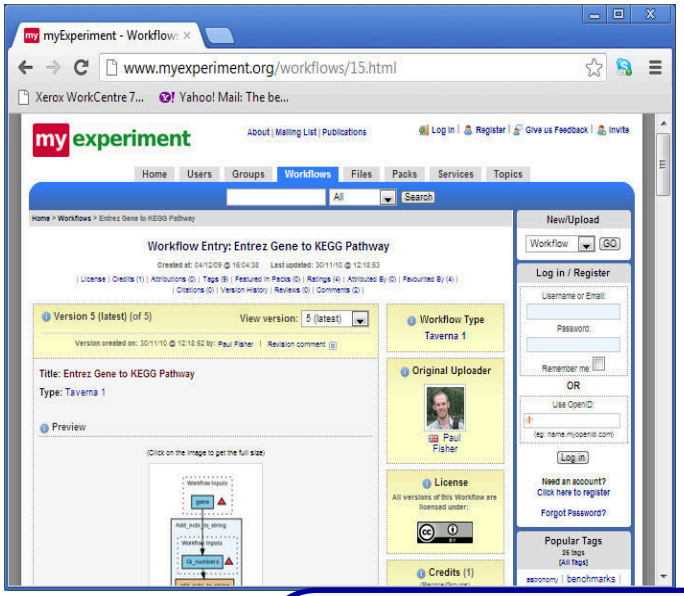
T. Oinn, et al. Bioinformatics, 2004.

Galaxy

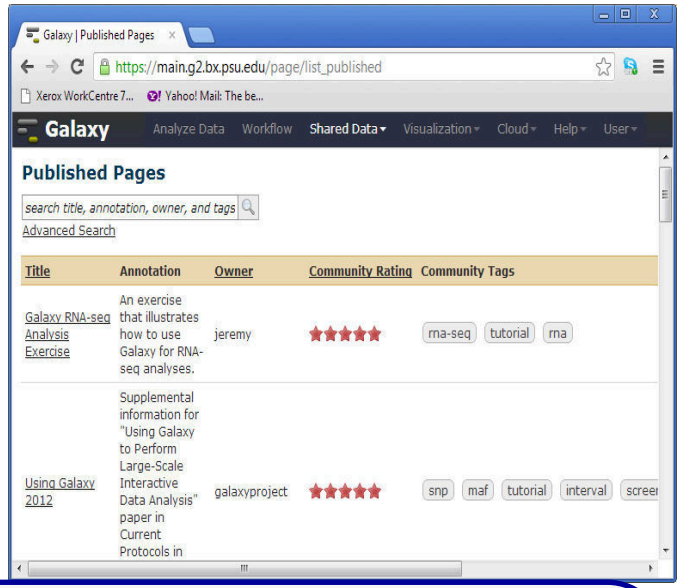
B. Giardine, et al. Genome Research, 2005.

Communities

Taverna MyExperiment

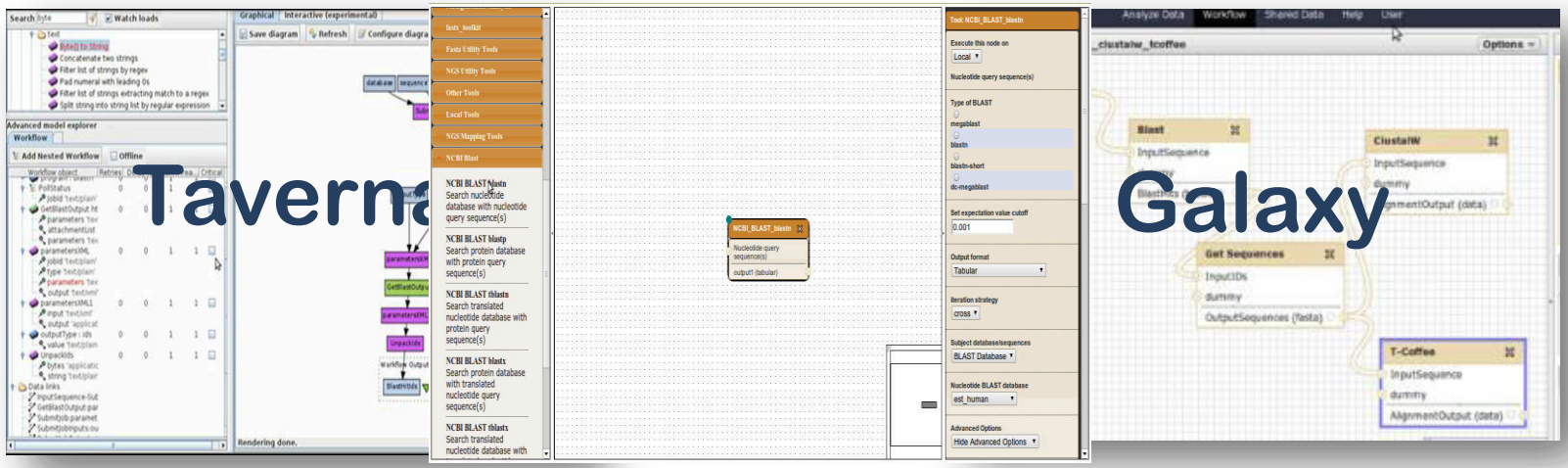


Galaxy Pages



What if we can make use of both repositories and what if we can have advantages of both systems??

Integrating Taverna and Galaxy Workflows



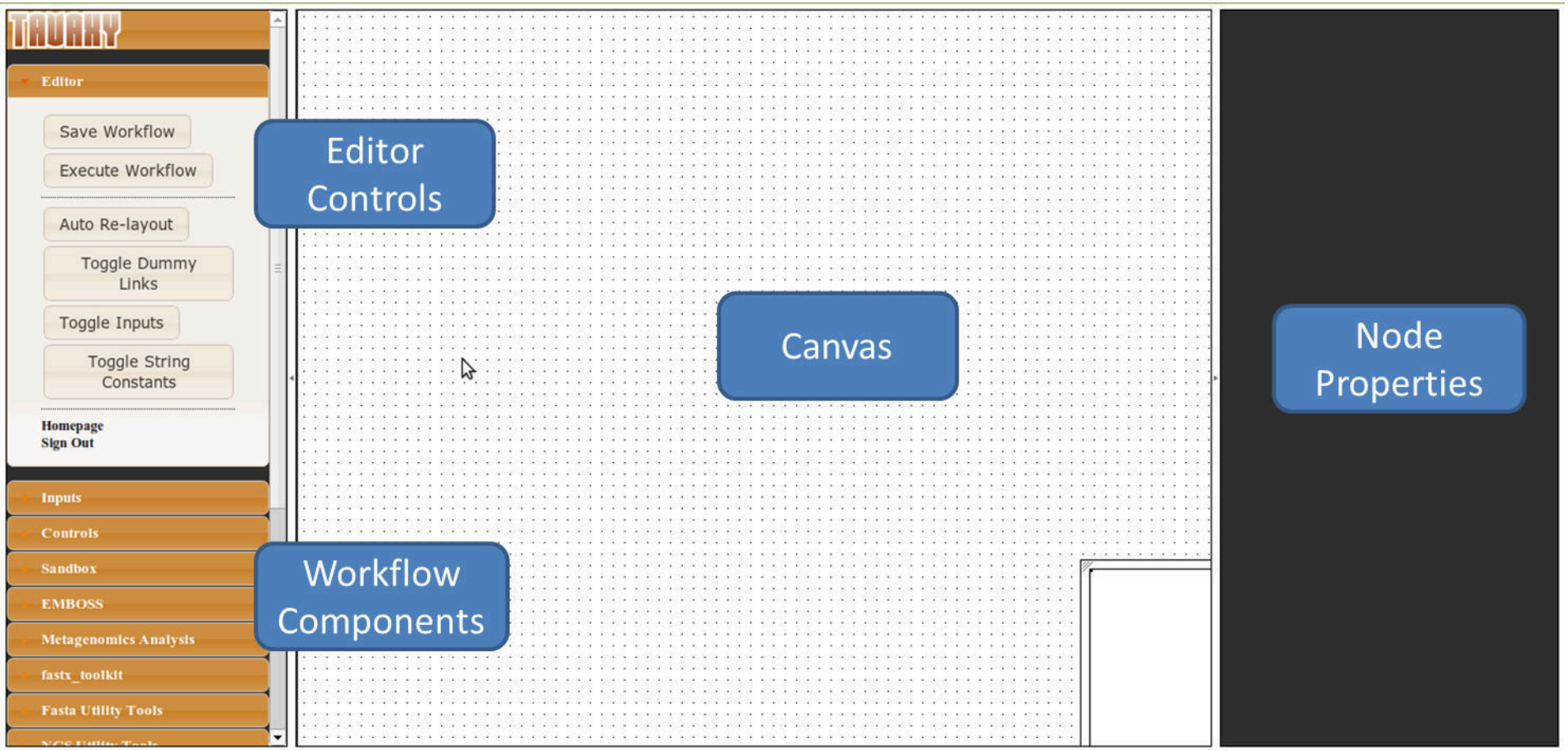
Integrating Taverna and Galaxy Workflows

Tavaxy

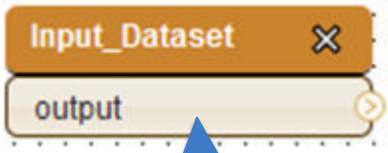
Tavaxy

- A standalone workflow system based on workflow patterns
- Integrates both Taverna and Galaxy workflows and improve their performance
- Easy to use and includes a large library of software tools
- Exploits high performance computing resources (parallel infrastructure) with all details being hidden
- Runs on local or cloud computing infrastructure
- Optimized to handle large datasets

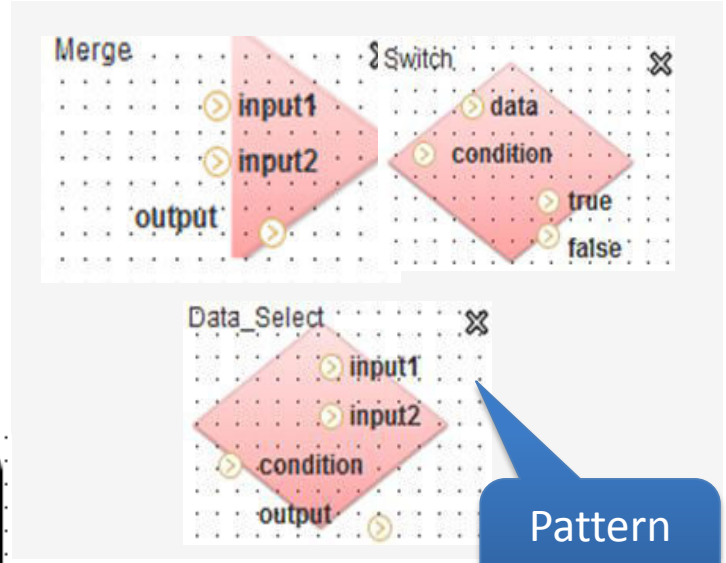
Tavaxy Environment



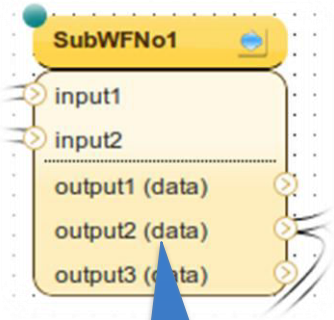
Tavaxy Nodes (Processing Units)



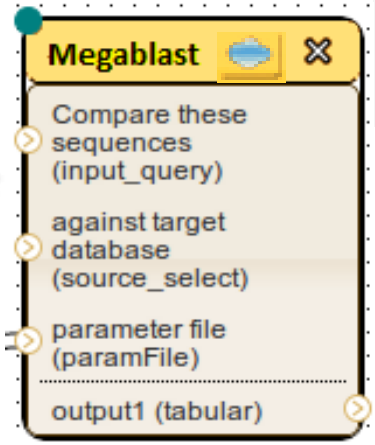
Input node



Pattern nodes



Sub-workflow nodes



Tool node

- If-else conditional
- Iteration
- Data Merge
- Data Select

New in Tavaxy and not in Galaxy

- Pattern nodes, Sub-workflow nodes
- Remote execution

New in Tavaxy and not in Taverna

- Easier to use interface
- Direct use of local HPC

Tools in Tavaxy

220 Tools organized in the following categories

- **EMBOSS:** Package of sequence analysis tools
- **SAMtools:** Package of scirpts and programs to handle NGS data
- **Fastx:** Package for manipulating fasta files

- **Galaxy tools:** A set of data utilities and tools developed by the galaxy team
- **Taverna tools:** A set of tools based on web-services based on Taverna. collection This section includes as well a set of data manipulation utilities developed by the taverna team.

- **Tavaxy tools**: This section includes extra utilities and tools for sequence analysis and genome comparison developed/added by the Tavaxy team.
- **Cloud utilities**: A set of data manipulation and configuration of computing infrastructure on the Amazon cloud.

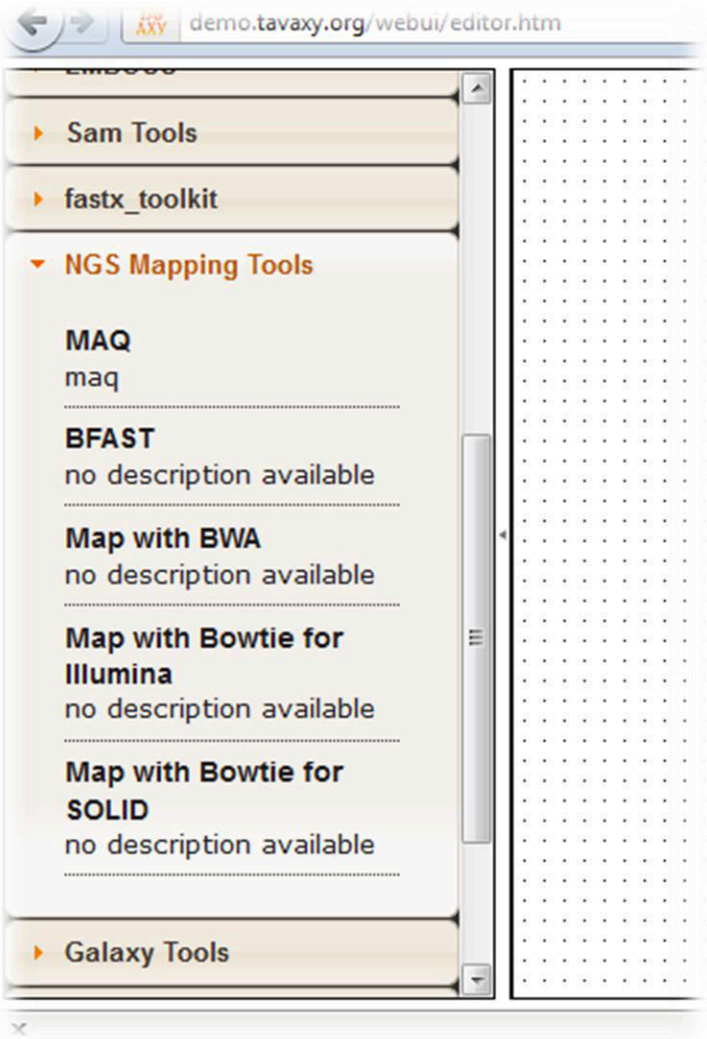
NGS Mapping Tools and Databases in Tavaxy

Tools

- MAQ
- BFAST
- BWA
- Bowtie
- …

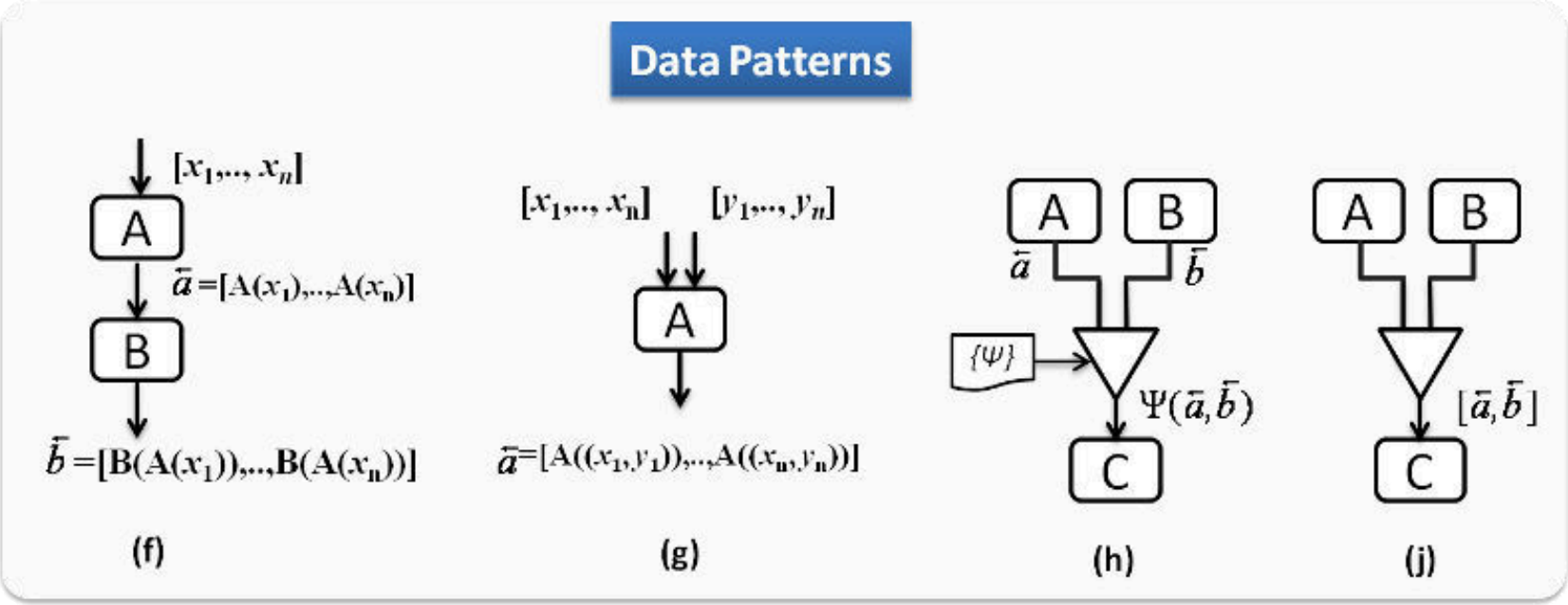
Databases

- Reference human genome hg36
- Indexed genome for MAQ, Bowtie
- NCBI nucleotide/protein DBs



Data Patterns of Tavaxy

To facilitate execution of data-intensive tasks in cloud computing cluster



Domenstration 1
Importing Taverna Workflow

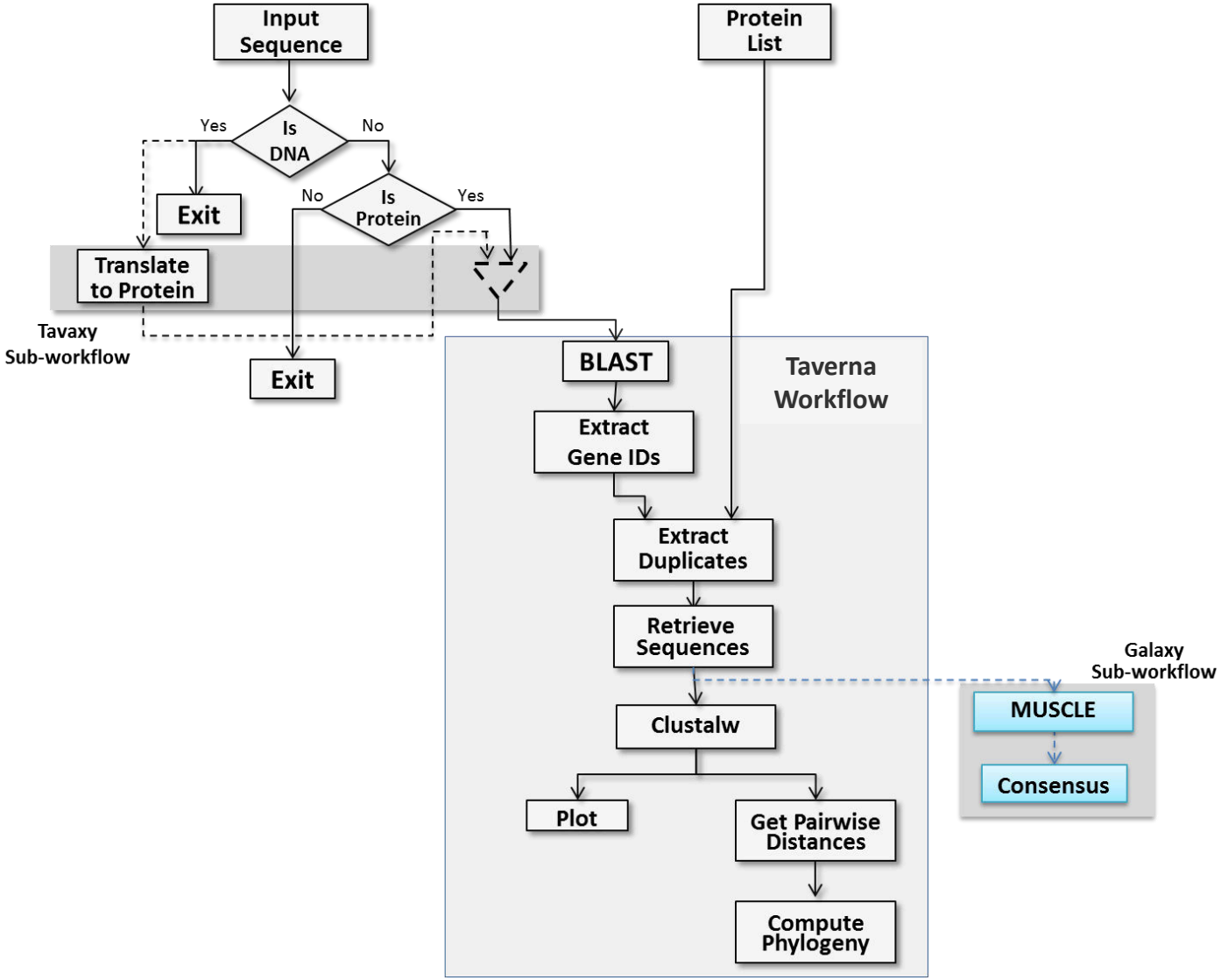
The *Tavaxy* Project

Single environment for integrating Galaxy and Taverna workflows:

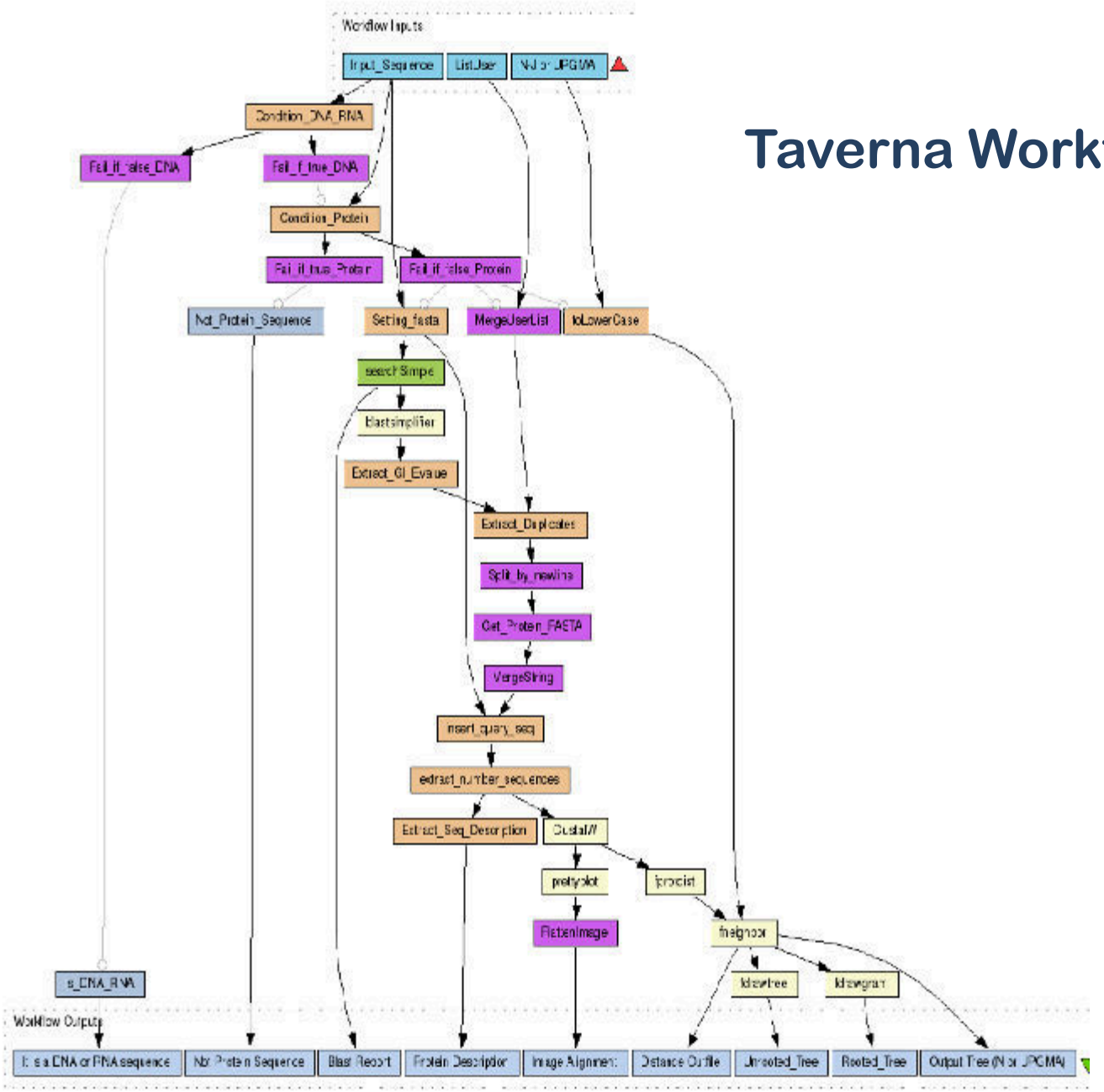
Taverna or Galaxy workflows can be

- imported , → open and re-draw in Tavaxy environment
- re-designed, → delete or re-order workflow nodes
- enhanced, → add new nodes and sub-workflows
- optimized,
  → Tavaxy concstructs can be used to exploit parallelization
  → remote Taverna calls can be replaced with local tools
- and executed in Tavaxy
  → on local (HPC) infrastructure
  → on cloud

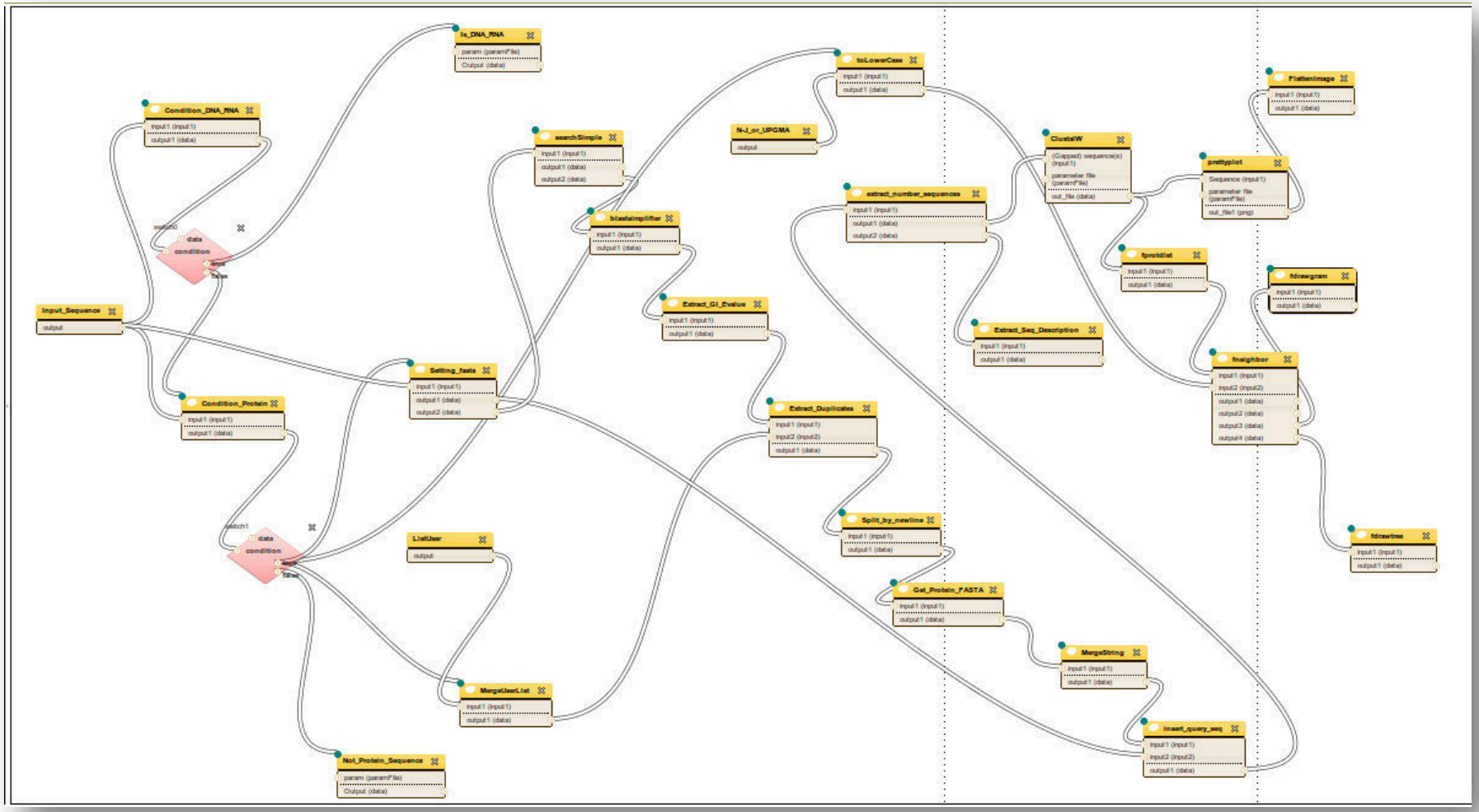
Importing Taverna Workflow



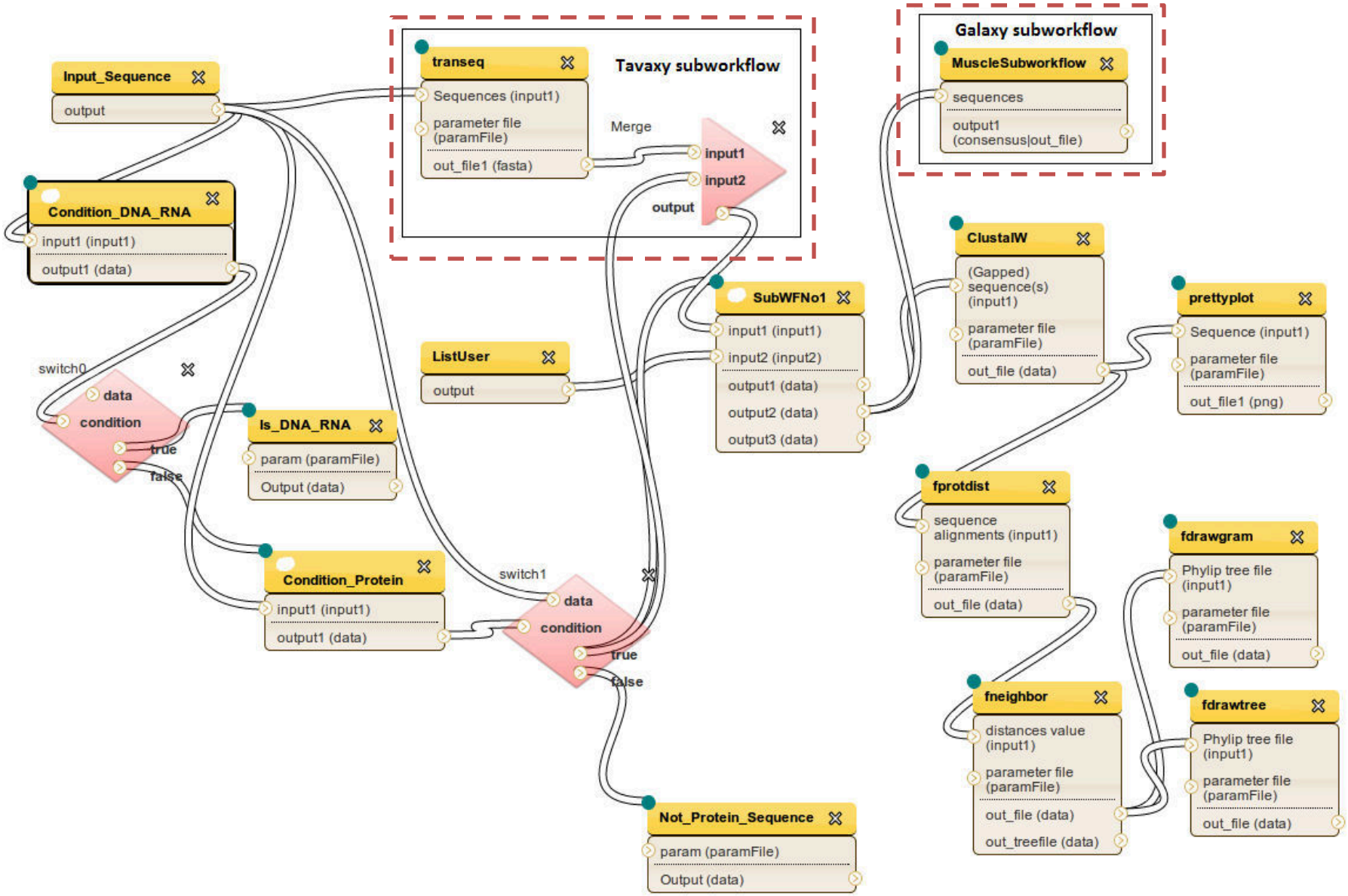
Taverna Workflow



Imported Taverna Workflow

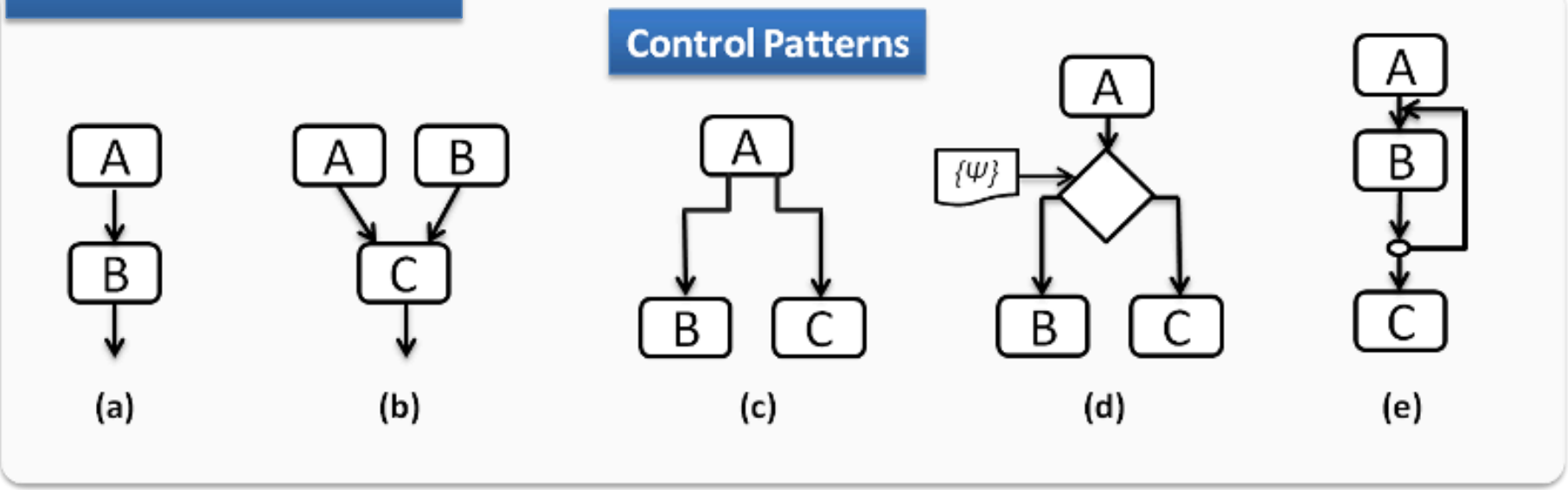


Optimized Imported Taverna Workflow



Idea of Integration

Workflow Patterns



Bag of techniques

- Taverna is used as a secondary engine to execute Taverna sub-workflows
- The use of patterns as special nodes in the data-flow oriented engine
- Simulating control constructs over the data flow digraph representing the workflow
- Use of sub-workflow to enable iteration pattern

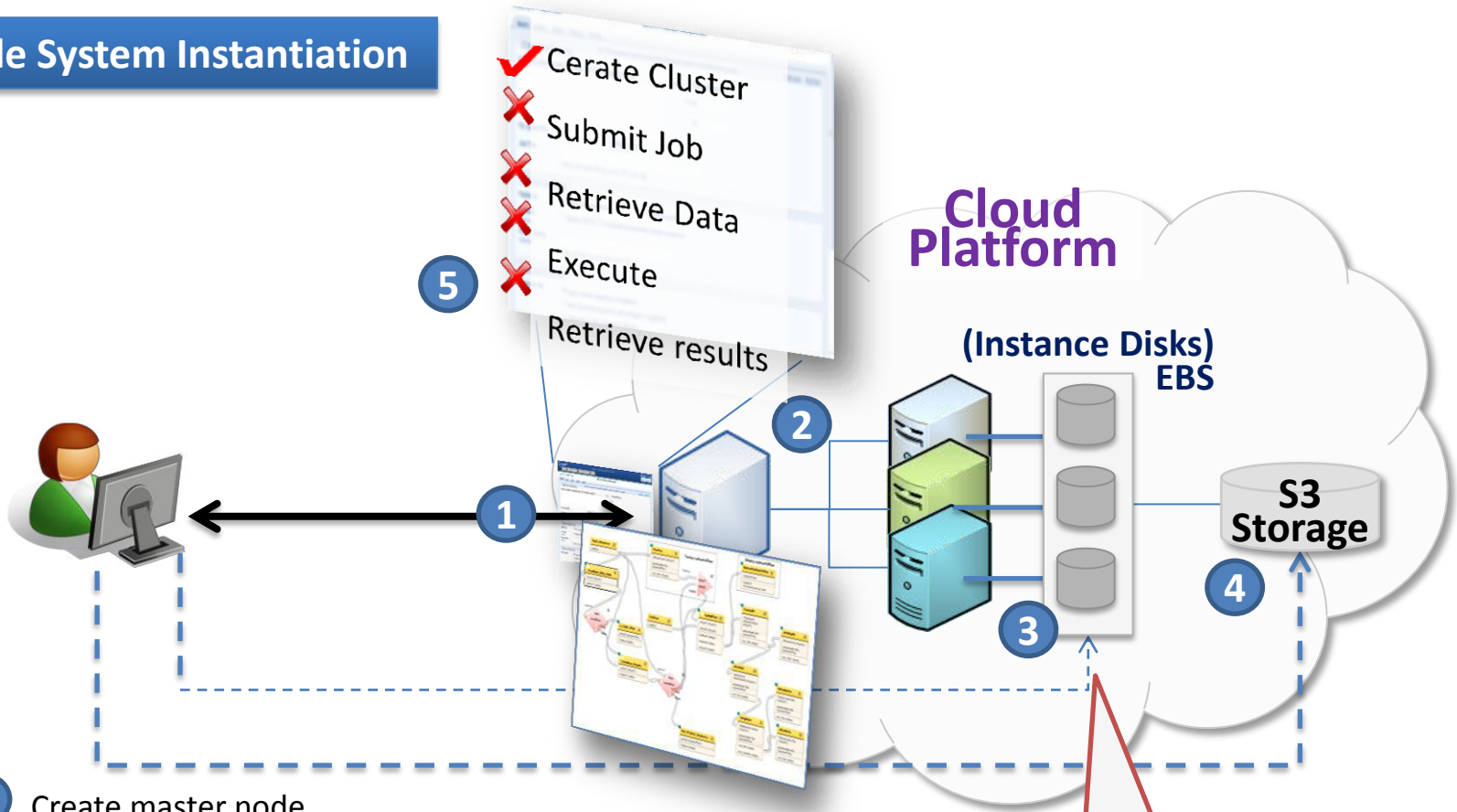
High Performance
Cloud Computing Support

Three Modes for Supporting Cloud

- Whole system instantiation
- Delegating execution of a sub-workflow to the cloud
- Delegating execution of one tool to the cloud

User case scenario

Whole System Instantiation

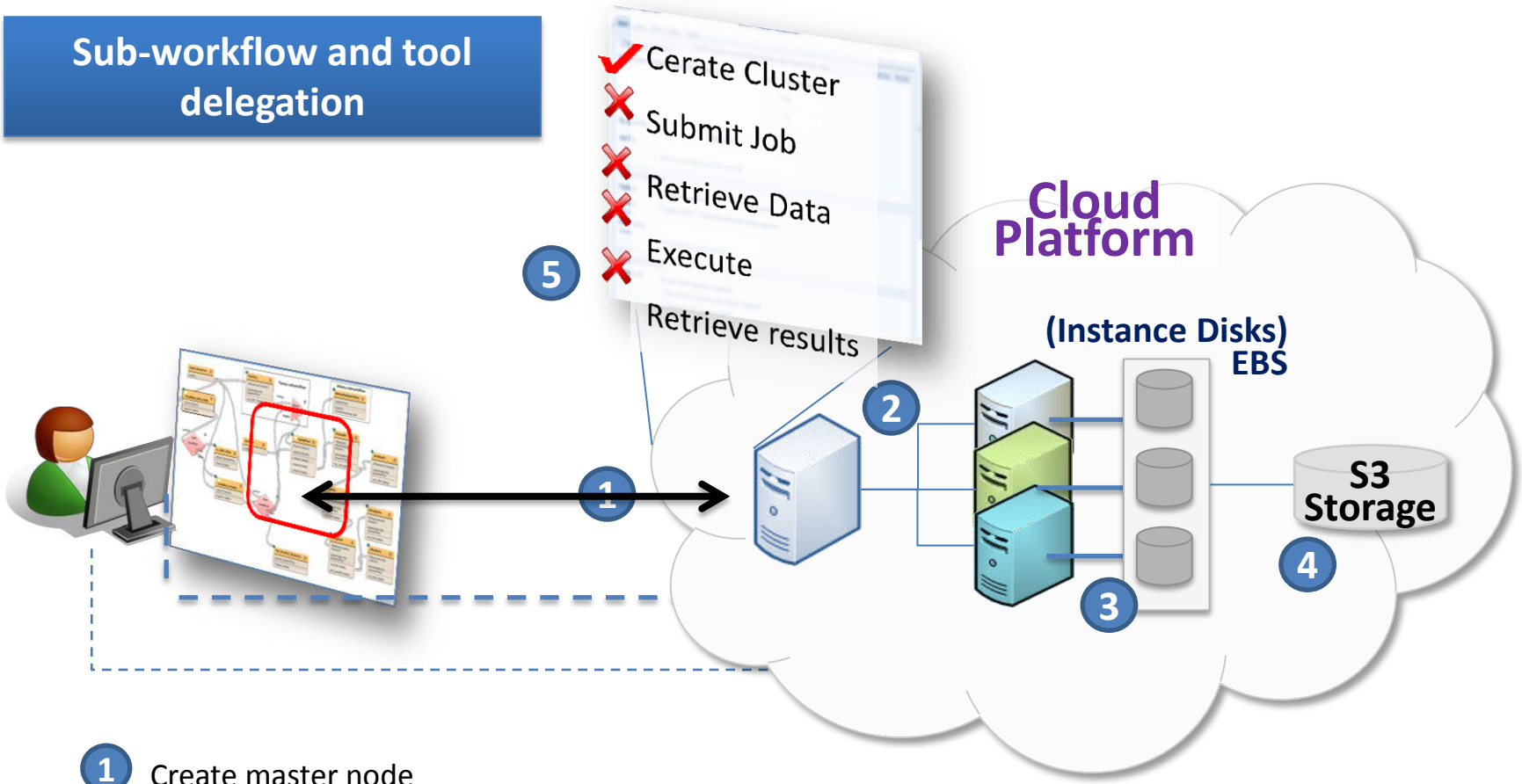


1. Create master node
2. Master node creates worker nodes
3. Instance disks are mounted
4. Connection to S3 is established
5. Submit data, command line (and executable) and execute

Include Tavaxy tools and DBs

User case scenario

Sub-workflow and tool delegation



1. Create master node
2. Master node creates worker nodes
3. Instance disks are mounted
4. Connection to S3 is established
5. Submit data, command line (and executable) and execute

Whole System Instantiation

About | Use Tavaxy | Help and Resources | Software Packages

I. Cloud-based Usage

The version of Tavaxy accessed from this page is hosted at the Amazon Cloud Computing platform. It does not run as a multi-user system installed on a certain local infrastructure. Rather, a separate infrastructure (composed of a computer cluster) with the whole Tavaxy system is created on the cloud for each user. That is, many users have many independent Tavaxy systems on many independent machines.

There are two options of using Tavaxy from this web-site

1. Use of free Tavaxy system; where we take over the cost of running that version. In this mode, the computation time is limited to two days and the cluster size is limited to 4 nodes.
2. Use of Tavaxy from another Amazon account; where the user takes over the cost of establishing the cluster and carrying out the computation. In this case, the user is required to enter his Amazon account information to proceed in the system. In this version, the user is free to set a cluster of any number of nodes and to let it run until computation is over.

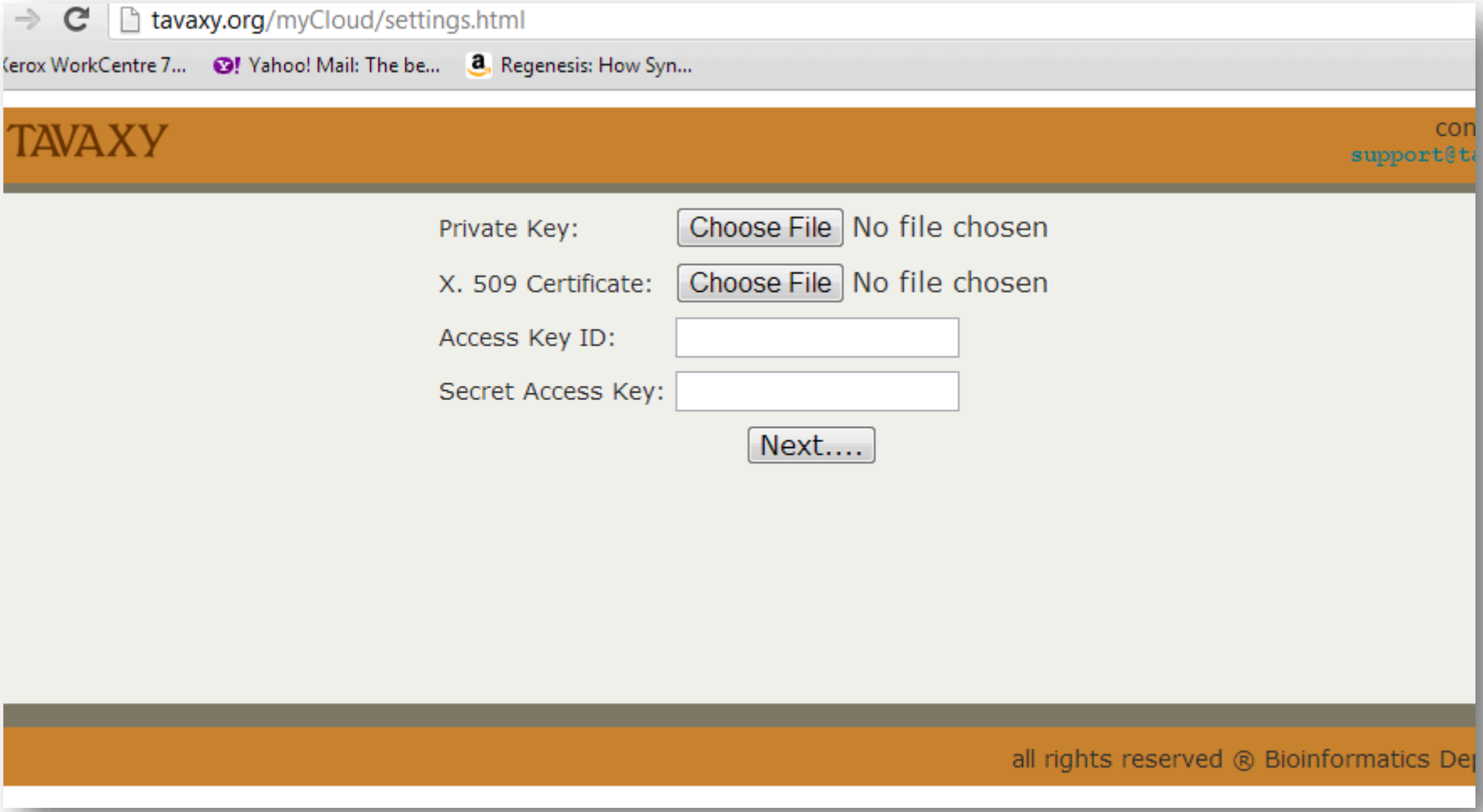
Start using Tavaxy on the cloud:

- Click here to start the free cloud version of Tavaxy (free)
- Click here to start Tavaxy from your cloud account (Pay-as-you-go)

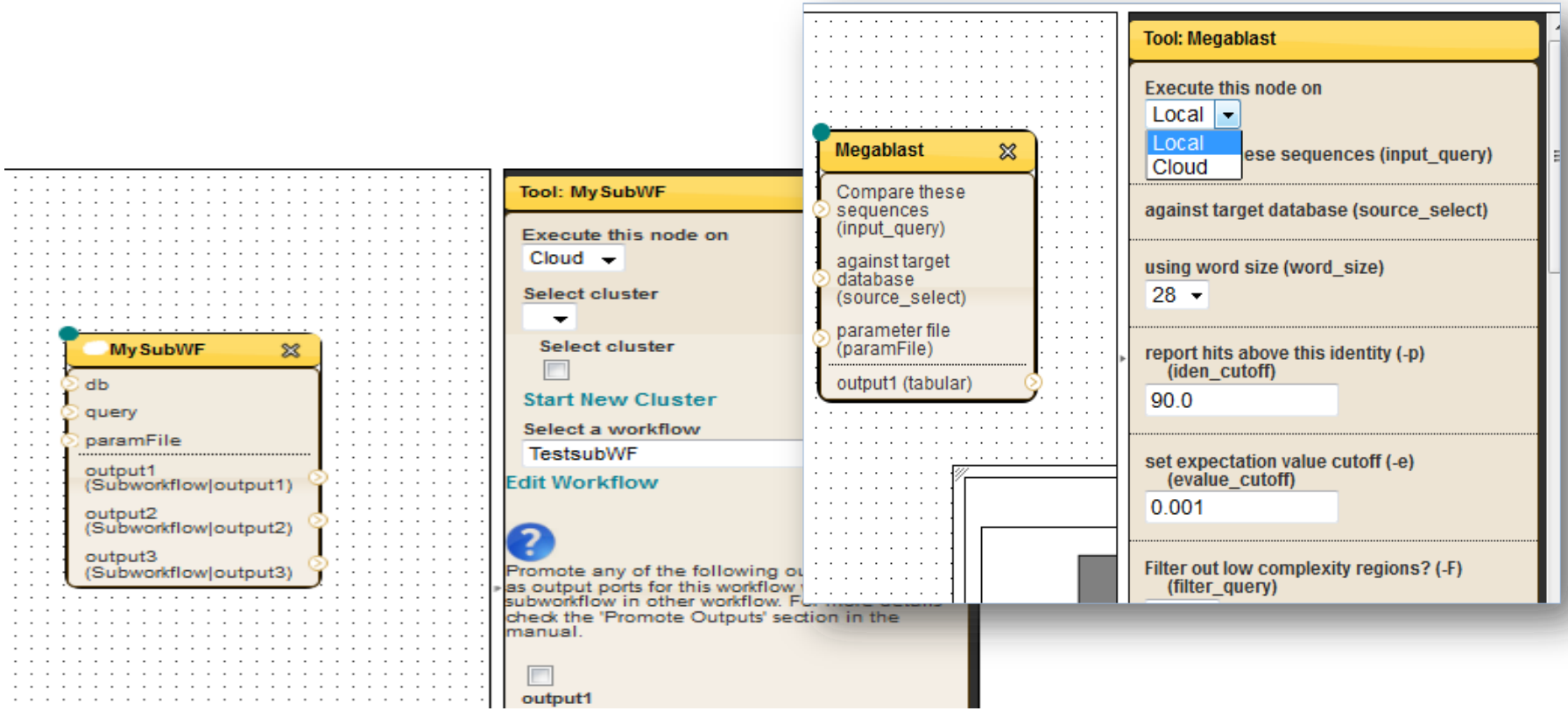
II. Web-server Usage

For small and medium jobs, you can use our demo server.
Demo Server: http://demo.tavaxy.org

Whole System Instantiation



Sub-workflow/Tool Instantiation

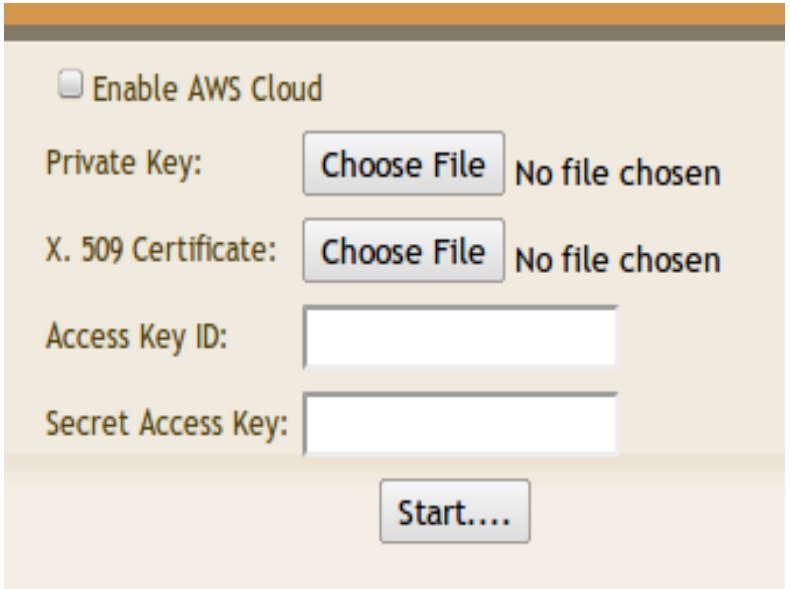


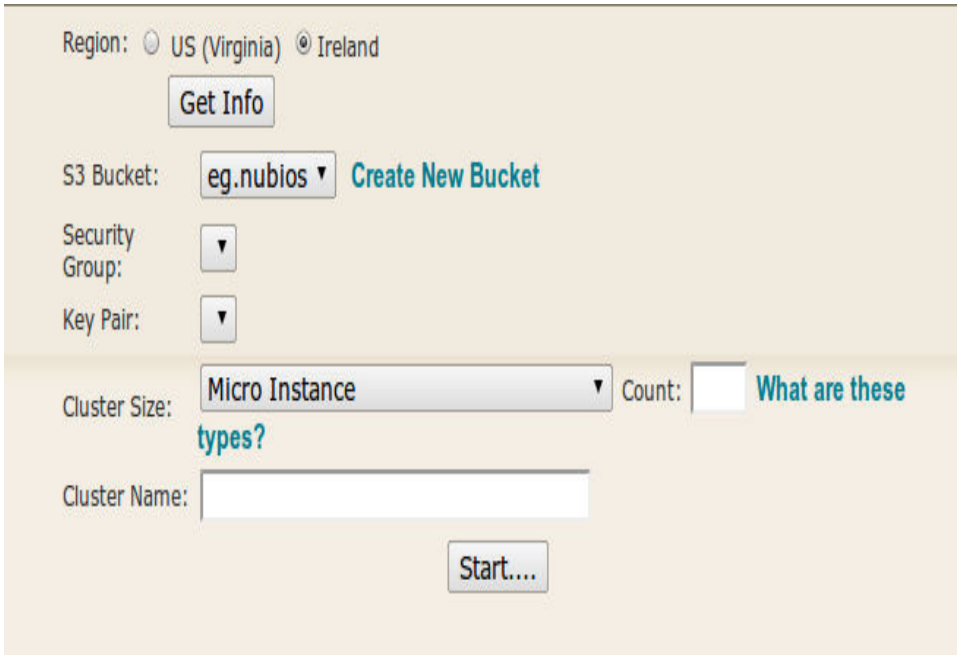
Exploiting Cloud Computing

Two steps:

1- Enter AWS credentials

2- Define your cluster

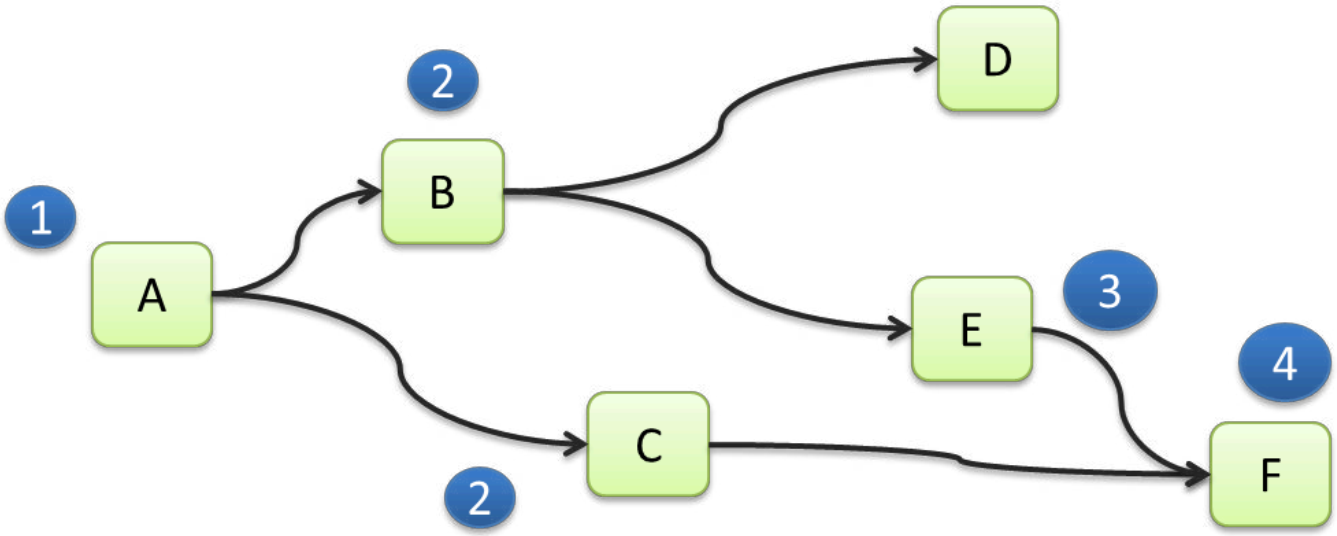




Supporting Task Parallelism

I. Parallelism due to branching

- Branching in the DAG implies independent execution
- Independent jobs can run in parallel

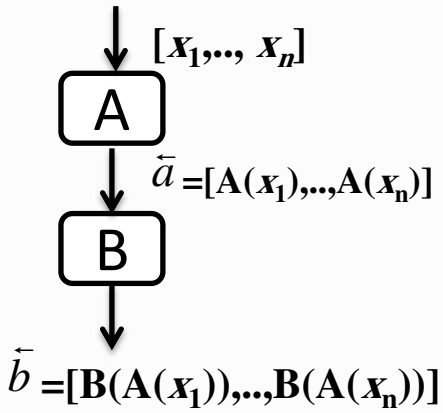


Supporting Data Intensive Tasks

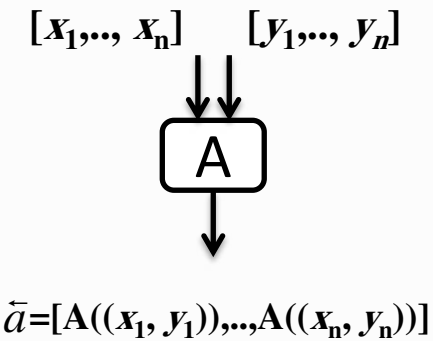
Data parallelism

- For an input as a list, node A can process the list items in parallel

Tavaxy Data Patterns



Single List



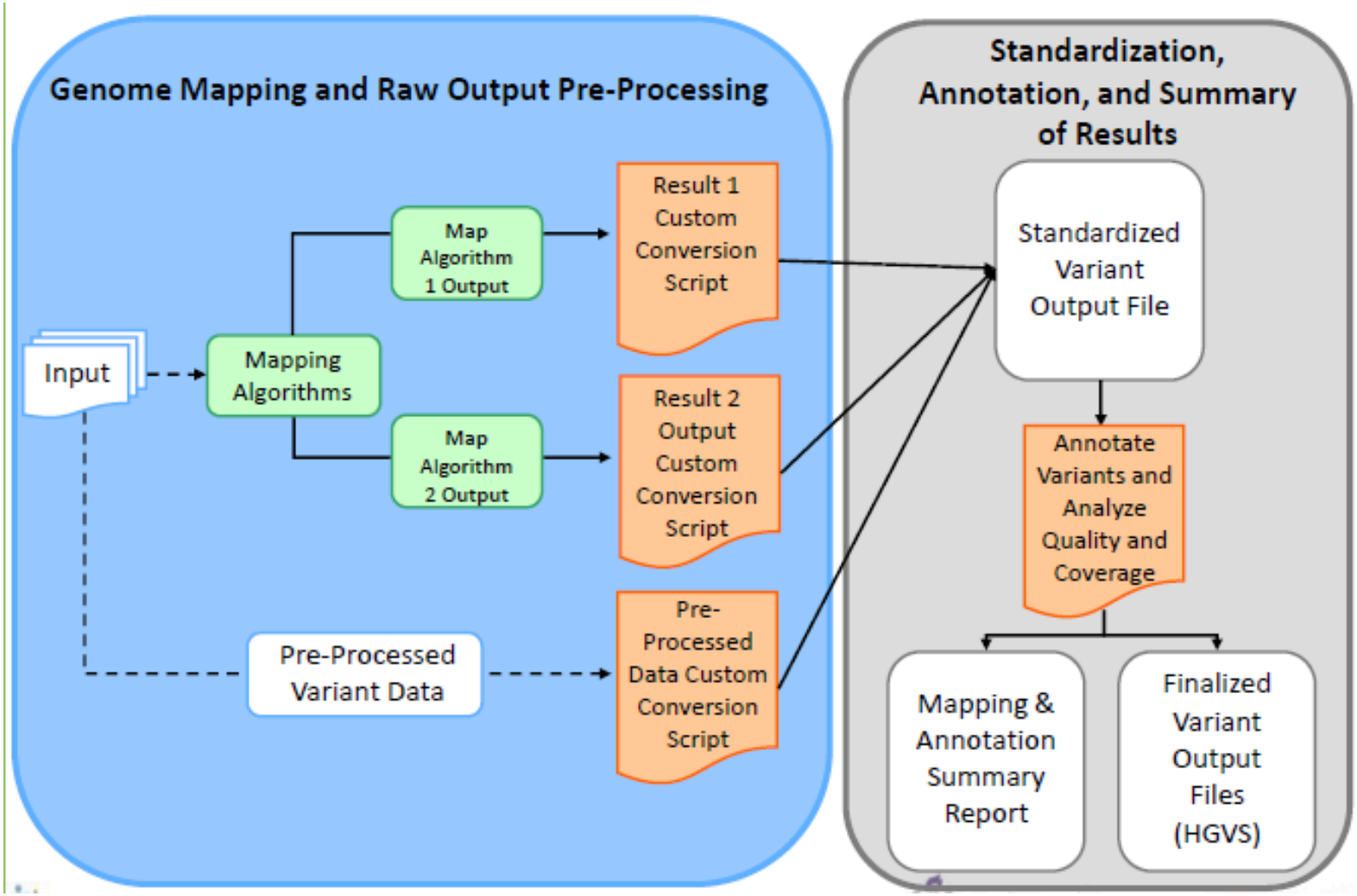
Multiple List

Personalized Medicine Workflow

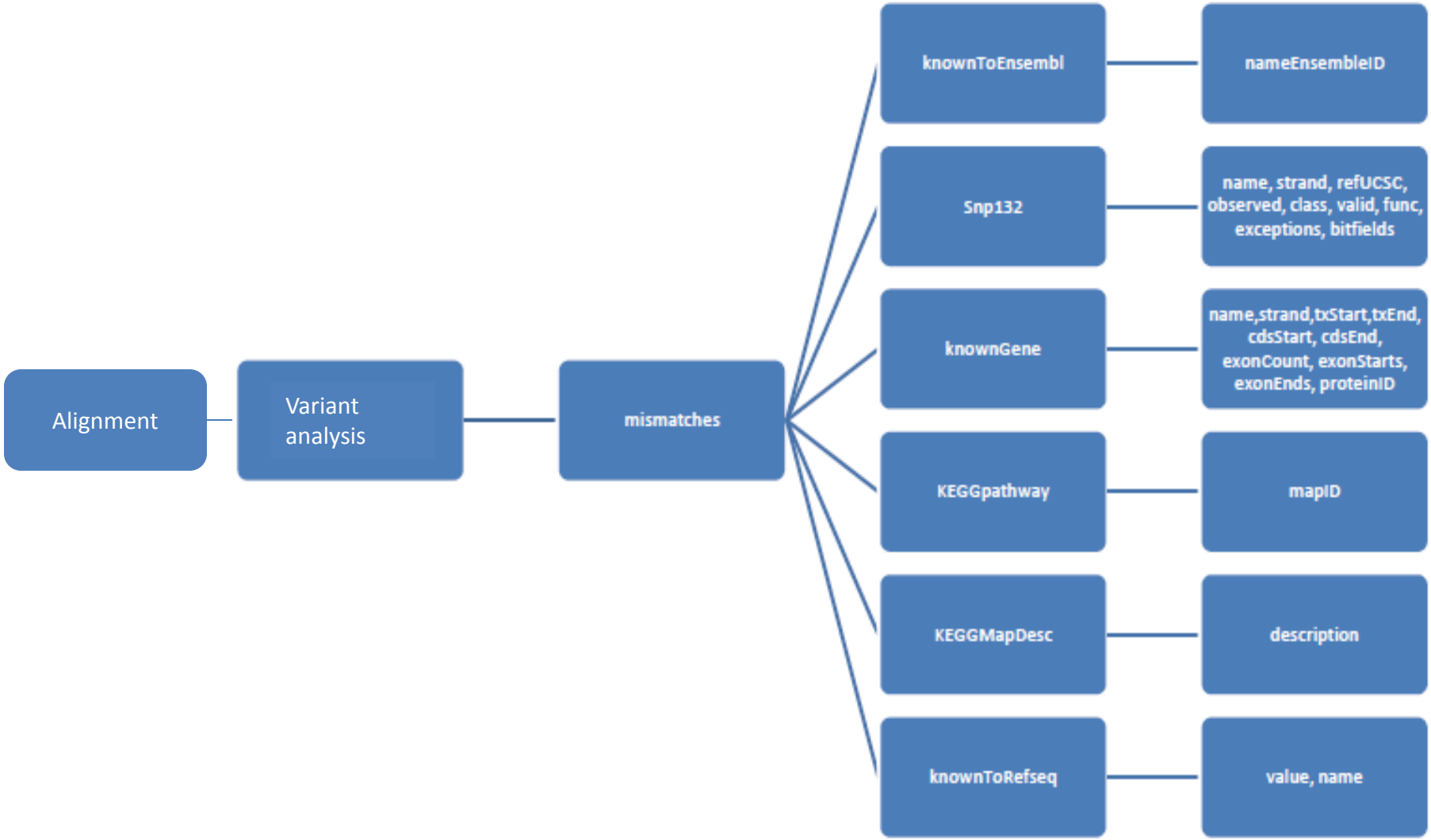
Personalized Medicine Workflow

Individual Whole Genome Mapping and Variant Annotation

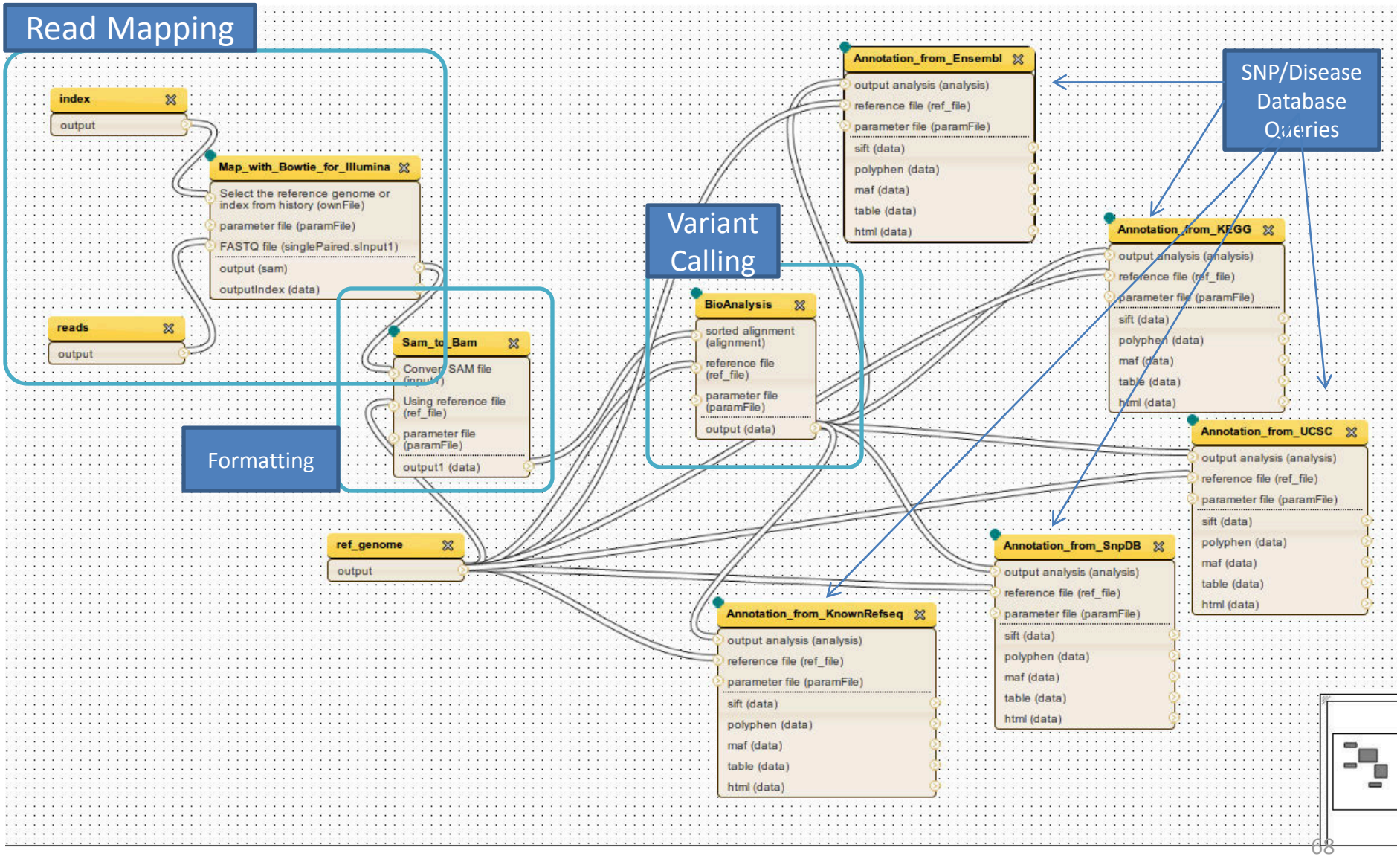
Tonellato-Wall



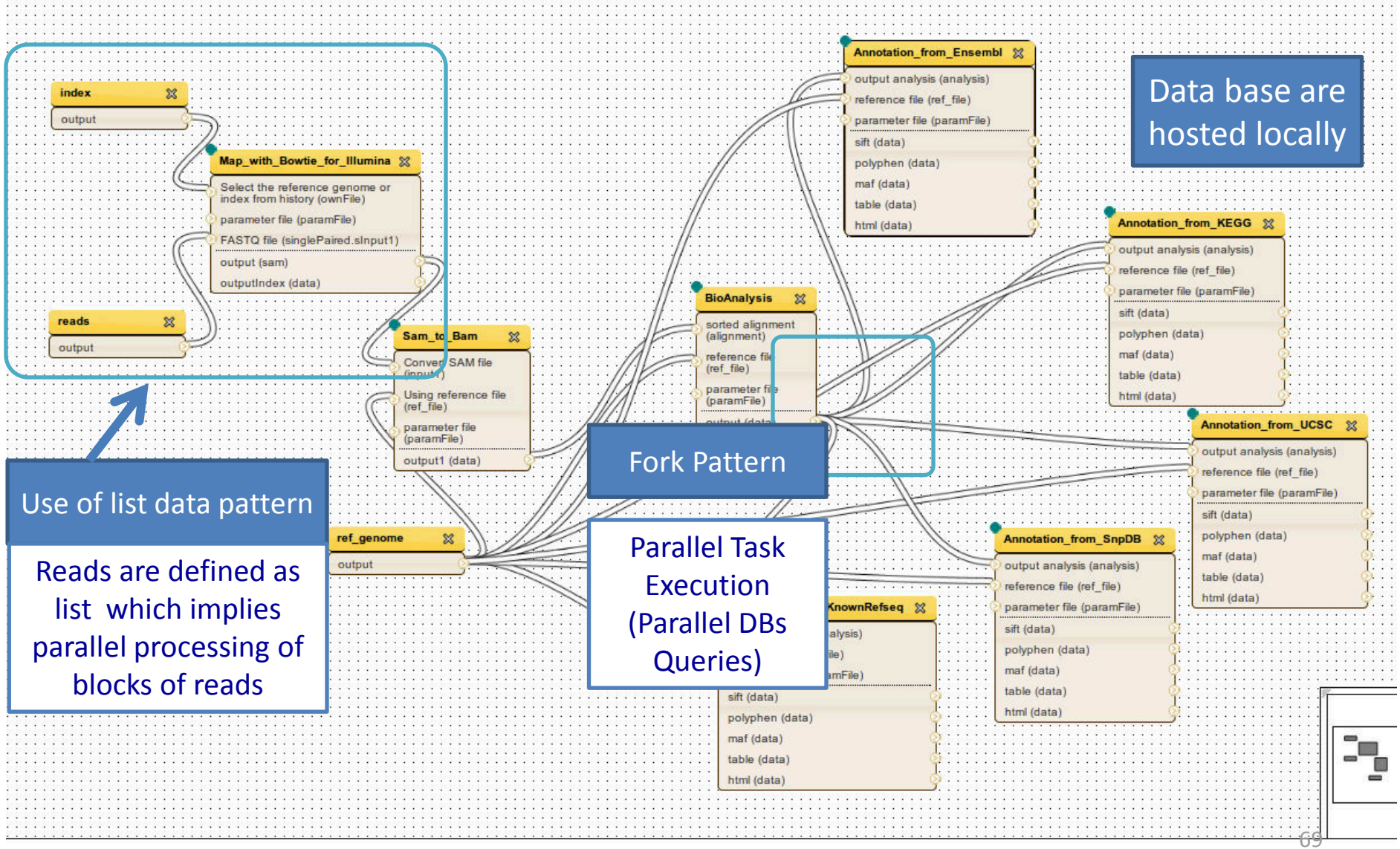
Workflow Sktech



Personalized Medicine Workflow on Tavaxy



Exploiting Parallelization and Locality of Data



Read-Mapping with Crossbow

- Crossbow (based on EMR/Hadoop) is used to map set of human reads to human genome.
- With this cluster, we mounted EBS volumes including the reference human genome files (each including one chromosome)
- Read Datasets:
  - illumnia reads of around 13 Gbp (47 GB) from from the African genome,
  - the human genome version hg18, build 36

Read-Mapping with Crossbow

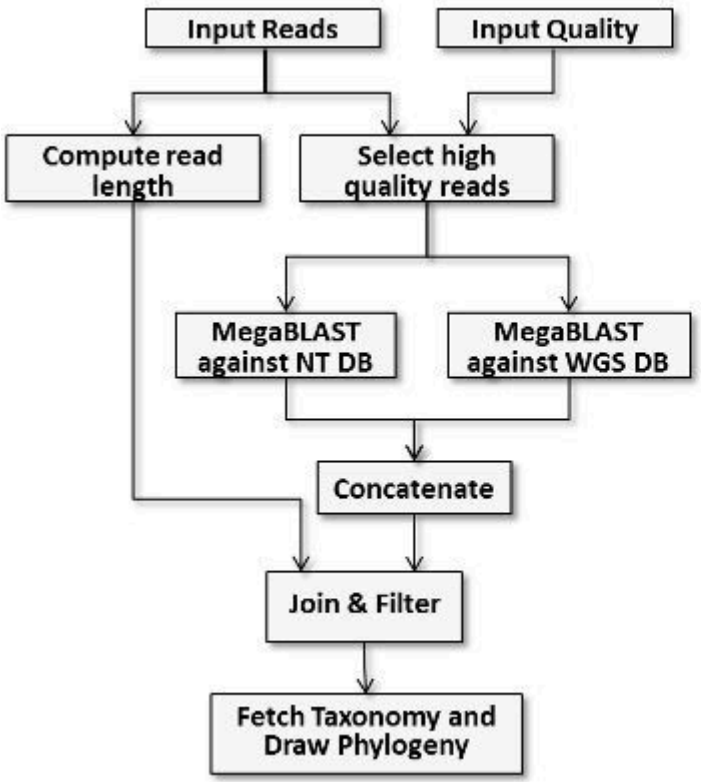
Table 4 - Running times of Crossbow on EMR using *elasticHPC*.

Num Nodes	Num Cores	Processing Time	Mapping Time	Total Time	Cost
			Using c1.xlarge		
1	8	66m	769m	835.1m	$1.68
4	32	39.5m	258.6m	298.4m	$2.4
8	64	35.25m	121.5m	156.8m	$2.88
16	128	34.1m	62.6m	96.9m	$3.84
24	192	33m	46.6m	79.8m	$5.76
32	256	33.0m	39.95m	73.5m	$7.68
64	512	32.65m	23.6m	56.1m	$7.68
			Using m1.xlarge		
1	4	72.2m	1675.6m	1748m	$2.7
4	16	40.6m	431.4m	472.6m	$2.88
8	32	37.3m	263.8m	301.1m	$4.32
16	64	33.6m	95.6m	129.6m	$4.32
24	96	32.9m	54.2m	87.1m	$4.32
32	128	32.6m	51.5m	84.3m	$5.76
64	256	32.8m	33.3m	66.1m	$11.52

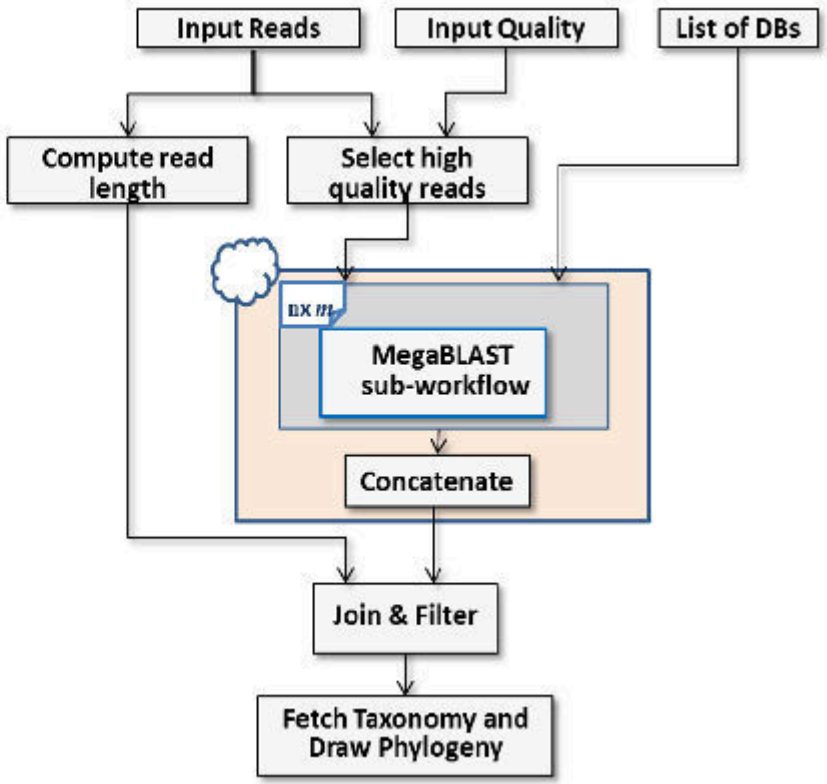
The average running times in minutes for EMR clusters of different sizes and machine types in the cloud. The machine types are c1.xlarge and m1.xlarge. A number in the column titled 'Total Time" is the summation of the pre-processing and alignment times.

Domenstration 3
Metagenomics Workflow

NGS-based Metagenomics Study

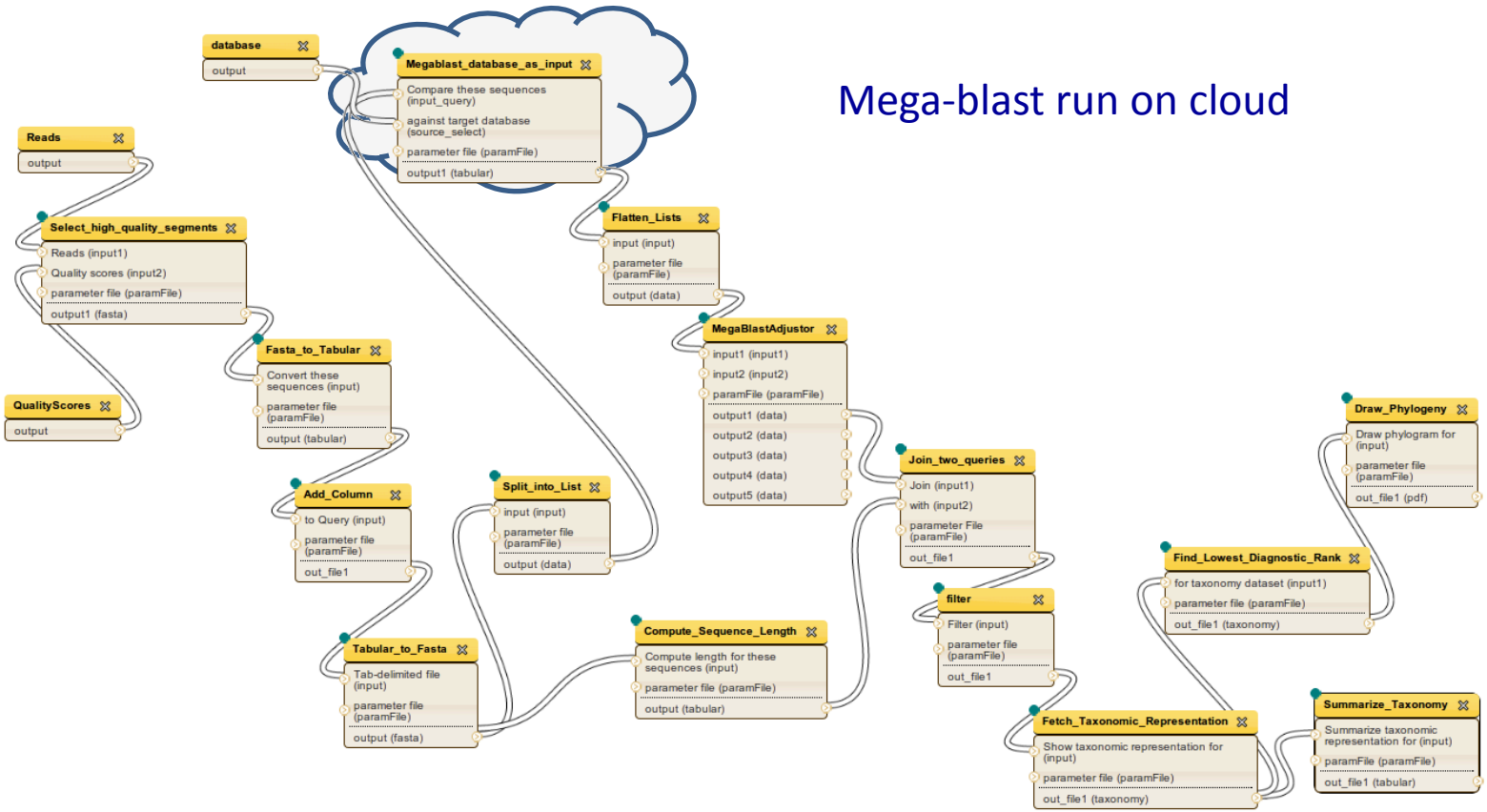


Galaxy Workflow



Optimized Tavaxy Workflow

NGS-based Metagenomics Study



Mega-blast run on cloud

NGS-based Metagenomics Study

- The MegaBLAST program is used to annotate a set of sequences coming from a metagenomics experiment.
- With this cluster, we mounted EBS volumes including the NCBI NT database
- Datasets: Windshield dataset composed of two collections of 454 FLX reads:
  - Trip A: 138575 (25.3 Mbp) Mb104283 (18.8 Mbp)
  - Trip B:) 151000 (30.2 Mbp) 79460 (12.7 Mbp)

Bioinformatics Experiment (2)

Dataset	AWS Cores				
	1	8	16	32	64
c1.xlarge (8 cores)					
	1 node	1 node	2 nodes	4 nodes	8 nodes
Trip A Left	93 ($1.32)	27 ($0.66)	20 ($1.32)	13 ($2.64)	9($5.28)
Trip A Right	127 ($1.98)	33 ($0.66)	21 ($1.32)	13 ($2.64)	7($5.28)
Trip B Left	80 ($1.32)	25 ($0.66)	17 ($1.32)	13 ($2.64)	7($5.28)
Trip B Right	65 ($1.32)	23 ($0.66)	13 ($1.32)	8 ($2.64)	6($5.28)
Total	365 ($4.62)	108 ($1.19)	71 ($2.64)	47 ($2.64)	29($5.28)
m1.xlarge (4 cores)					
	1 node	2 nodes	4 nodes	8 nodes	16 nodes
Trip A Left	77 ($1.28)	18 ($0.64)	13 ($2.64)	9 ($5.12)	7($10.24)
Trip A Right	119 ($1.28)	34 ($0.64)	25 ($2.64)	16 ($5.12)	10($10.24)
Trip B Left	70 ($1.28)	31 ($0.64)	23 ($2.64)	15 ($5.12)	9($10.24)
Trip B Right	65 ($1.28)	27 ($0.64)	13 ($2.64)	9 ($5.12)	6($10.24)
Total	331 ($2.56)	110 ($1.28)	74 ($5.12)	49 ($5.12)	32($10.24)

The average running times in minutes for traditional computer clusters of different sizes and machine types in the cloud. The machine types are c1.xlarge and m1.xlarge. The numbers in brackets are the computation costs in US Dollar for the US-East site with $0.66 per hour for c1.xlarge and $0.64 per hour for m1.xlarge. (Note that partial computing hour of an instance is billed on Amazon as a full hour) The cost in the column titled "Total Time" is not the summation of the above rows, but it is the cost of the total running time if the four datasets in the respective column were processed altogether in the cluster.

Conclusions and Future Work

- Use of workflow systems provides flexibility and efficiency
- High performance and cloud computing resources are exploited with technical details being hidden
- Future work include
  - Further performance optimization for execution on local and cloud infrastructures.
  - Supporting multiple cloud providers
  - Handling larger data sizes in multi-use environment

Thanks for attention