Specifying *In Silico* Experiments as Coordinations of coarse-grained Processes

Ane Tröger

Department of Computer Science

University of Manchester

Introduction – ISXL

- Support in silico experiments through a specification language
- **I**n **S**ilico e**X**periments **L**anguage - ISXL
- ISXL facets
  - Constructs of the domain as DSLs (few constructs, but powerful ones)
  - High level specification (coarse-grained processes)
  - Methodological principles (implicit enforcement)
  - Support for long-lived investigations (persistent language)

Introduction – Non-programmed move from *in vitro* to *in silico* practices

- Movement from in vitro to in silico sciences, mainly in Bio
- Significant results in data generation and data integration
- Some results in process co-ordination as well
- Workflow in Bioinformatics
- Workflows on their own do not incorporate the methodological principles that experiments should (and could) have
- In science, experiments – in e-science, in silico experiments

THE UNIVERSITY
of MANCHESTER

Introduction – Non-programmed move from *in vitro* to *in silico* practices

- Biological data is, in essence, heterogeneous, autonomous and distributed;
- Efforts have been devoted to data semantics
- Efforts devoted to process coordination benefits from work done in data semantics;
- The grain of process coordination is too fine: the concentration of efforts might have been much too concentrated in data integration, “pure” process coordination is lags behind

ISXL – Two aspects

Specifying ***In Silico* Experiments** as Coordinations of **Coarse-grained Processes**

- What do we consider to be *in silico* experiments?
- What do mean by coarse-grained processes?

One example of an *in silico* experiment

- Distant Homology
- Aim: to infer homology by successive comparisons between sequence and profiles generated by multiple alignment

Distant Homology Protocol

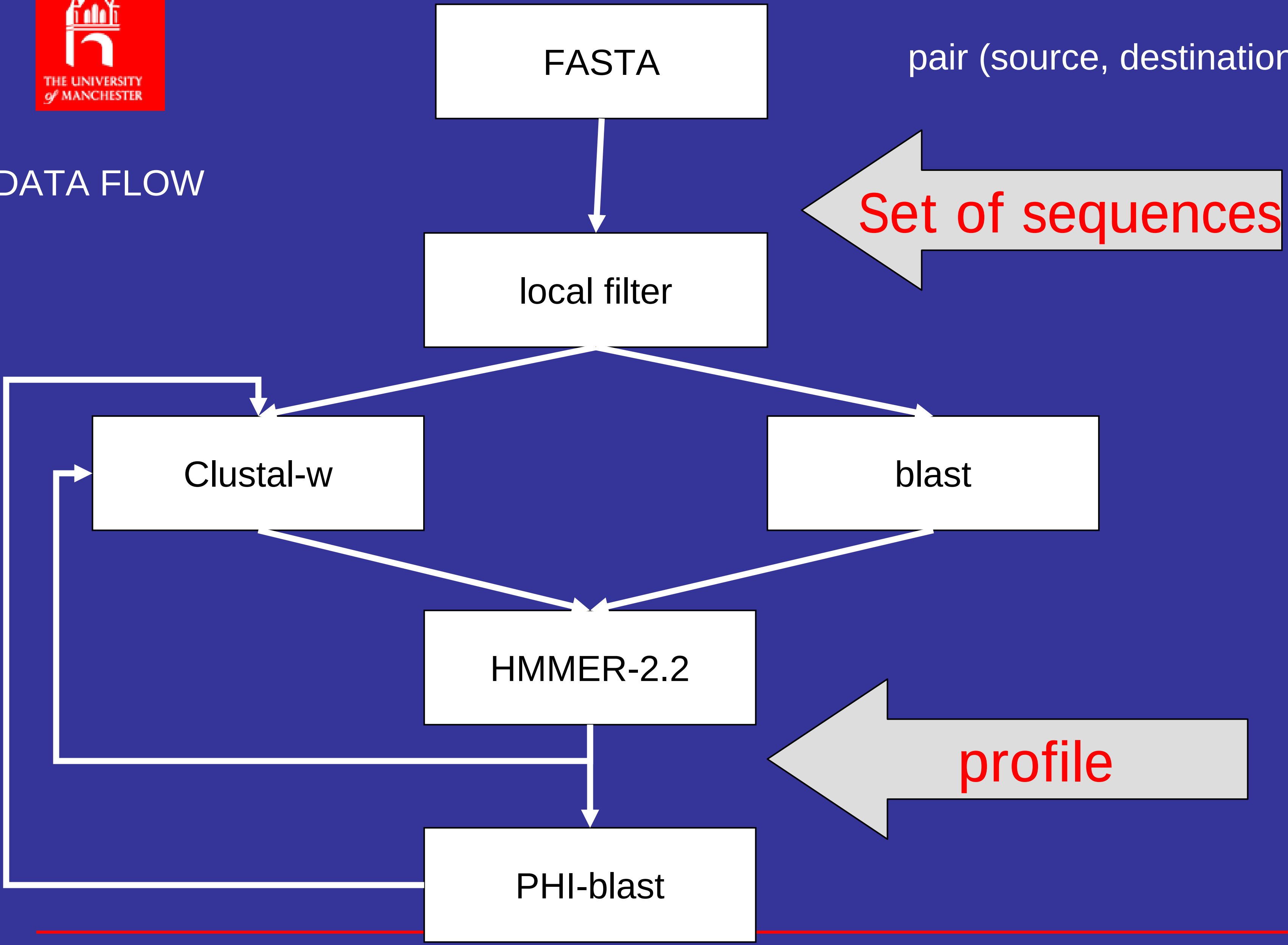
1. Given a sequence, retrieve close homologues
2. Filter G+C content(60)
3. If there is more than one sequence after (2), perform multiple alignment
4. With sequence in (1) and alignment in (3), generate profile
5. Retrieve more sequences with profile
6. If no more homologues are found, the last profile and sequences are the most distant related
7. If there are more sequences, a new profile is generated (4)

Coordination Aspects

- Distant Homology involves:
- different data format and granularities (items or sets): sequences, multiple alignments and profiles generated and related sequences used and produced by
- different tools: searches, filtering, alignment and profile generation and comparisons that require
- integration and coordination: data and control flows

DATA FLOW

pair (source, destination)



FASTA

Set of sequences

local filter

Clustal-w

blast

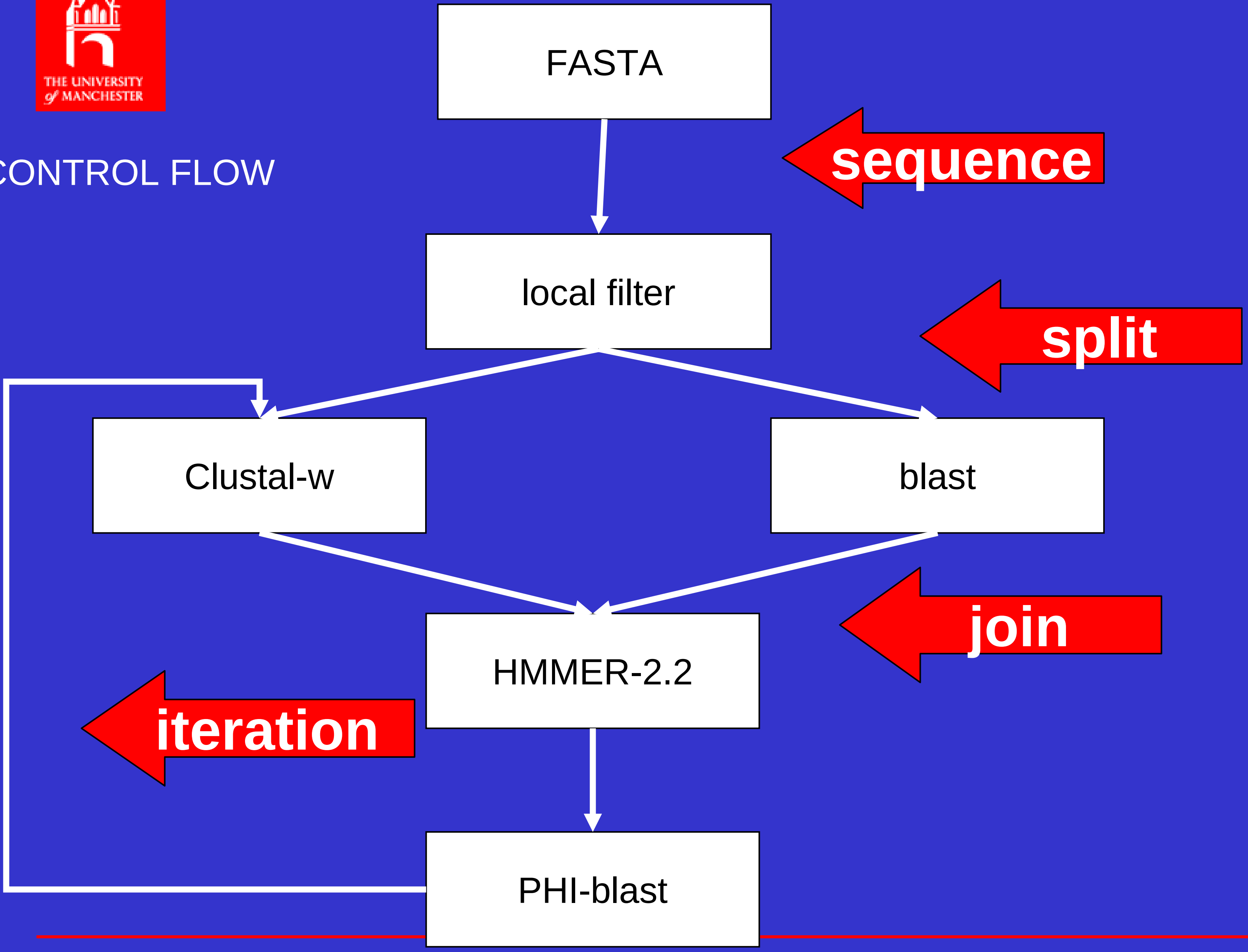
HMMER-2.2

profile

PHI-blast

THE UNIVERSITY of MANCHESTER

CONTROL FLOW



Coordination Aspects

- **Data flow is characterized by the data transfer from source to destination, taking into consideration HAD resources**
- **Control flow is characterized by the dependencies between tasks: conditions determine sequence, splits, joins and iterations**

Coordination Aspects

- **Raise the level of data types and functions to the level of services**
- **Wrappers are particularly suitable to abstract away from the semantic level**
- **Wrappers for data containers and processes can be stateful**
- **Data and control flow situations can be captured by task states and data container states**
- **These map to materials and methods and protocol, so as to use constructs of the domain: experiments**
- **The grain gets coarsen to the level of the coordination**

Coordination Aspects

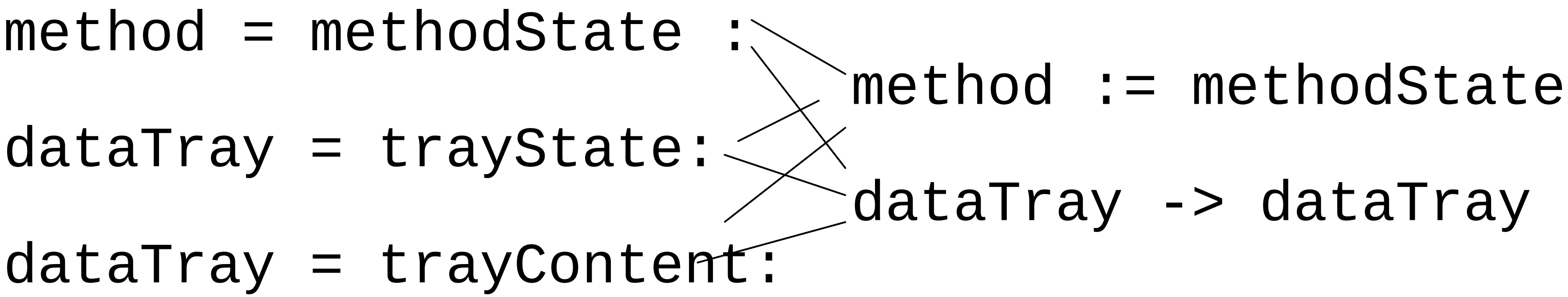
- To reach the semantic levels, tasks are web services, composition of ISXL tasks or EVAL calls with the appropriate bindings to data objects
- Bindings are paired up by the compiler (one end is ISXL, the other, values in (preferably) interpreted languages (for example Python, or BioPython)

ISXL Syntax and Semantics

- (production) rules determine the combination of processes and data
- Constructions of the domain are materials for data and methods for processes
- Both are stateful, meaning that, the monitoring and control can be more detailed, specially if control and data flows are separate (not dependent) constraints
- These are the operators of the language

- RETE algorithm
- `Given a set of rules;`
- `Select the ones that can be triggered (condition evaluate to true);`
- `That yields a conflict set;`
- `Apply a conflict resolution policy (order);`
- `Fire the first rule;`
- `Repeat until there are no rules in the conflict set.`

Expressiveness: control and data flows can be detached



method = methodState :

dataTray = trayState:

dataTray = trayContent:

method := methodState

dataTray -> dataTray

AND – OR - NOT

What is the difference from other workflows for bioinformatics?

- Why would anyone want to know about distant homology?
- Because conjectures are part of science...
- Formulation of hypothesis

- This is the point where workflows are not enough anymore:they have not, so far, been enriched with methodological constraints

- Hypothesis are relationships
- Relationships are functions

- Functions can become computations

- In ISXL, functions are processes, which are represented as protocols

- Hypothesis can be considered a type of protocol, reasoning on the evidence gathered

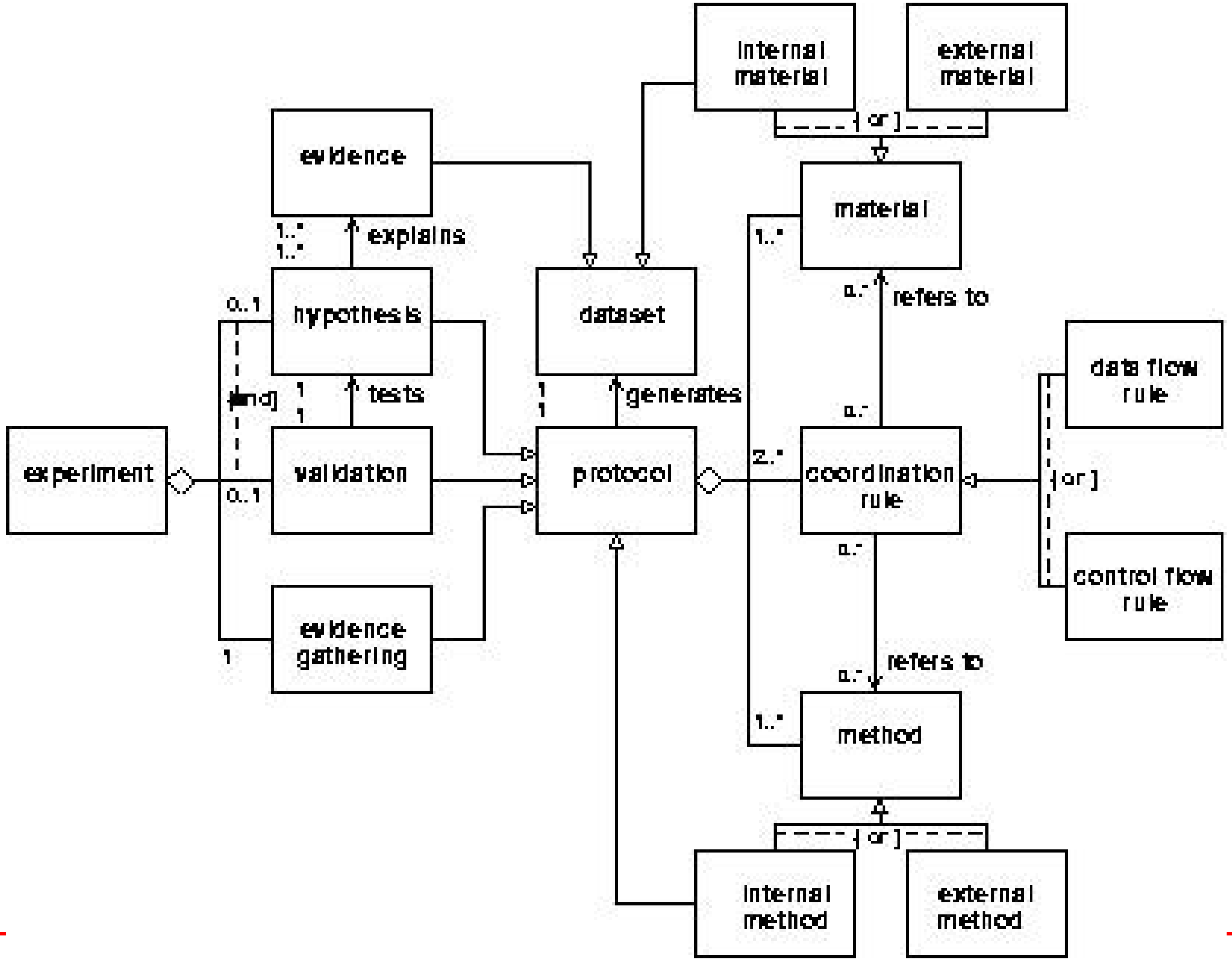
What is the difference from other workflows for bioinformatics?

- Hypotheses acquire strength or weakness when statistical measurements are attached to it
- Statistical measurements, or the validation procedure, can also be seen as functions...
- validation procedure is a protocol that test the relationship established by the hypothesis as many times as there are evidence gathered

This relationship can be implicitly enforced by the following (nested) function:

V(H(E(input sequence)

ISXL Conceptual Model



What is the difference from other workflows for bioinformatics?

- Experiments can be interrelated
- Experiments can evolve, giving rise to lineages, families of experiments
- Keeping the information of this conceptual model as metadata each time an experiment is specified and conducted enriches the language with memory (persistence mechanism)
- This gives support for long-lived investigations
- This is already another story...

Conclusions

- Most bioinformatics tools for process co-ordination only provide:
  - A computational model of the evidence gathering stage of the experimental method
  - The means to specify coordination of processes in experiments that involve semantic types (focusing away from the coordination requirements)
- With ISXL, scientific practices in silico, however, involve relating:
  - The evidence gathered with an explicit hypothesis and an explicit validation process (what we call the methodological principles);
  - Complex topologies for workflows at coordination level, which is orthogonal to the semantic complexity and heterogeneous nature (this is why the specification at coarse-grained is relevant)

Specifying *In Silico* Experiments as Coordinations of Coarse-grained Processes

http://www.cs.man.ac.uk/~trogera

http://www.cs.man.ac.uk/~alvaro

Ane Tröger

Department of Computer Science

University of Manchester

ISXL Syntax and Semantics

- Hypothesis, evidence gathering and validation are processes
- In ISXL, processes are protocols. A protocol consists of separate control and data flow rules
- The main data abstraction is the notion of materials that flow into and out of trays
- The main procedural abstraction is the notion of methods that associate tasks to in- and out-trays
- Data flow rules specify the conditions under the state of in- and out-trays change
- Control flow rules specify the conditions under which the state of method changes
- ISXL refrains from having built-ins
- Procedural primitives are accessible in an underlying language through a meta-call to `eval`
- Data primitives are (roughly) streamed files, e.g., those denotable by URLS