Newcastle University

ReComp: optimising the re-execution of analytics pipelines in response to changes in the data

Paolo Missier
School of Computing
Newcastle University, UK

1st CMLS workshop @ER 2020
November 4, 2020

EPSRC
Engineering and Physical Sciences Research Council

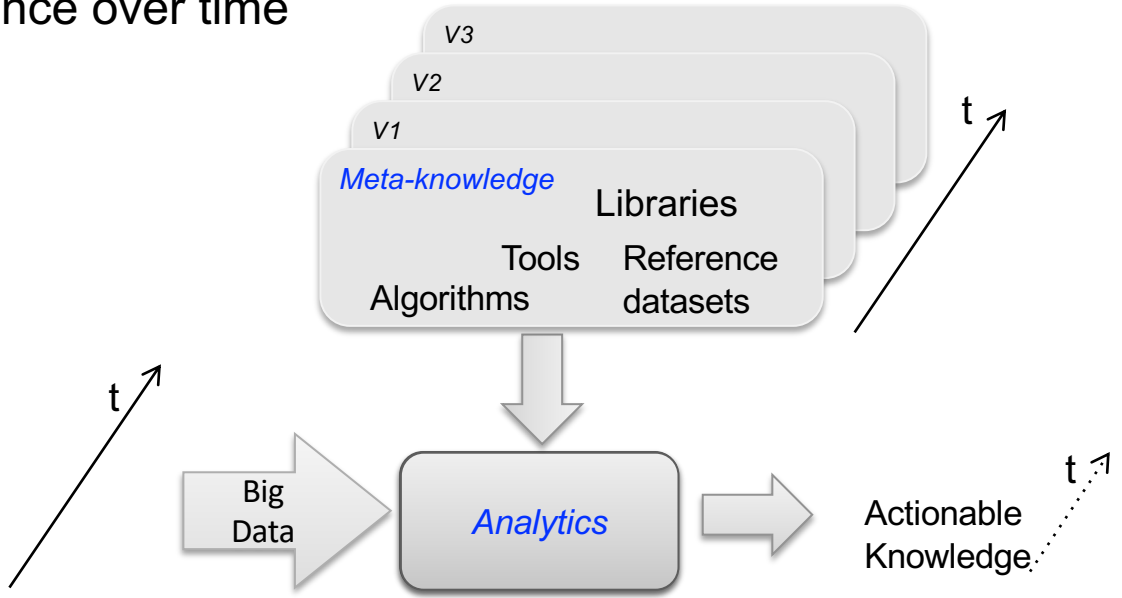
Topics

Data evolution and its impact on data-driven processes

The ProvONE provenance model and role in selective process re-run

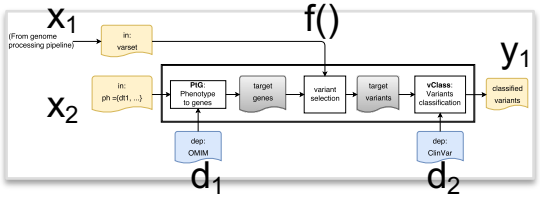
ReComp outlook: from source queries to analytics outcomes

Data Science over time



Reacting to changes in inputs

$$f : X \rightarrow Y \qquad \mathbf{x} = [x_1 \dots x_n] \qquad \mathbf{y} = f(\mathbf{x})$$
$$\mathbf{y} = [y_1 \dots y_m]$$



Possible reactions to changes in any of the inputs:

$$\mathbf{x} \rightsquigarrow \mathbf{x}'$$

1. Always refresh

simply compute $\mathbf{y}' = f(\mathbf{x}')$

inefficient if computing $f(.)$ is expensive, and

y, y' turn out to be very similar to each other

2. Approximate

find a new function $f'(.)$ that approximates $f(.)$

return $f'(\mathbf{x}')$

Define a distance metric δ_Y on Y

try and estimate $\delta_Y(y, y')$ *without explicitly computing* y'

if $\delta_Y(y, y') > \Delta_Y$ for a set threshold Δ_Y then compute $f(\mathbf{x}')$

This approach works well when:

1. Distance metrics can be defined on both X and Y: δ_X, δ_Y
2. $f(.)$ is *stable*: $\delta_X(\mathbf{x}, \mathbf{x}') < \epsilon_X \Rightarrow \delta_Y(f(\mathbf{x}), f(\mathbf{x}')) < \epsilon_Y$

If these assumptions hold, then simply:

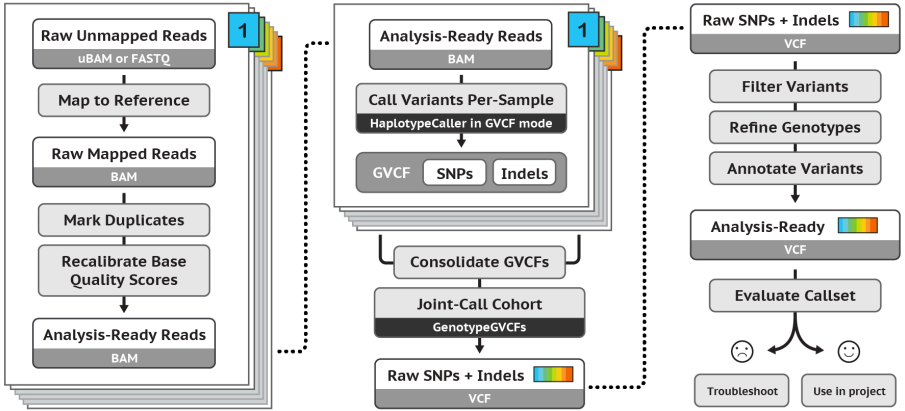
Compute $\delta_X(\mathbf{x}, \mathbf{x}')$ and use it to *estimate* ϵ_Y

ReComp iff $\epsilon_Y > \Delta_Y$

Case study: genomics

Image credits: Broad Institute https://software.broadinstitute.org/gatk/

Variant calling / interpretation

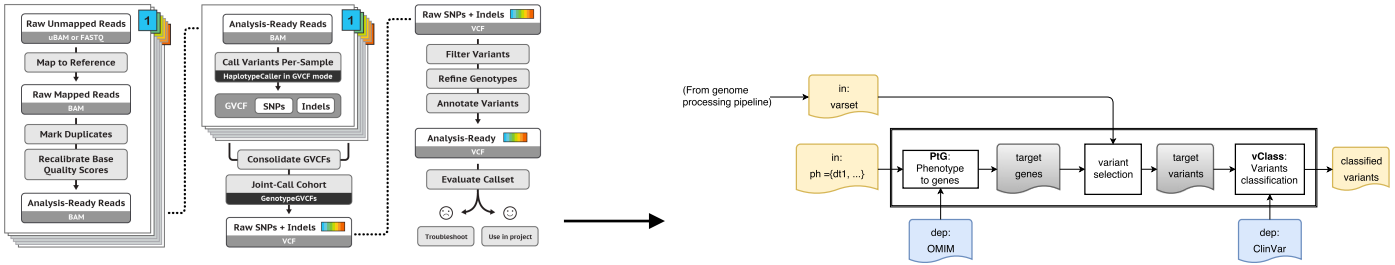


Spark GATK tools on Azure:
45 mins / GB
@ 13GB / exome: about 10 hours

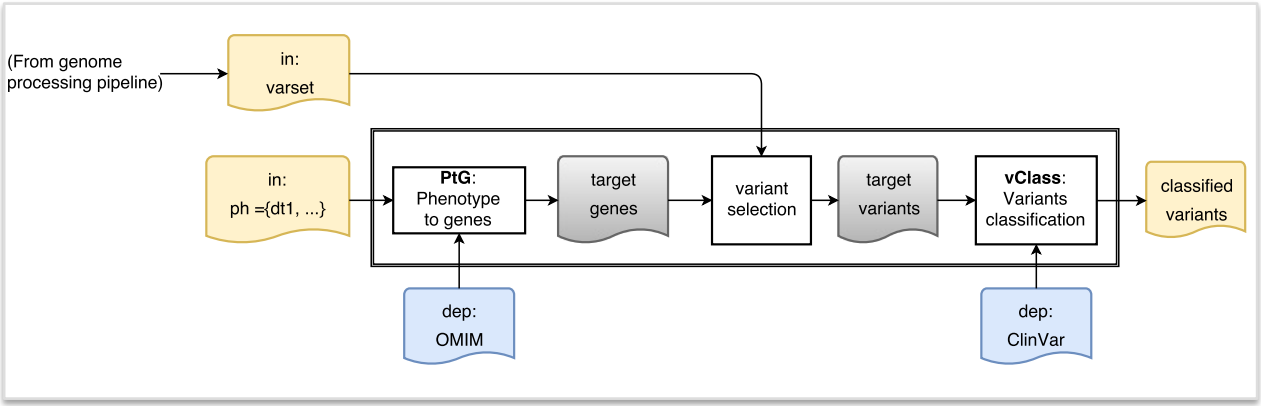
Genomics england

https://www.genomicsengland.co.uk/the-100000-genomes-project/

Genomics: WES / WGS, Variant calling → Variant interpretation



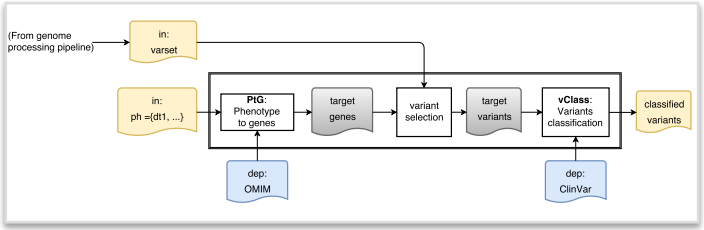
Variant classification : pathogenic, benign and unknown/uncertain



SVI: a simple single-nucleotide Human Variant Interpretation tool for Clinical Use. Missier, P.; Wijaya, E.; Kirby, R.; and Keogh, M. In *Procs. 11th International conference on Data Integration in the Life Sciences*, Los Angeles, CA, 2015. Springer

Reacting to changes in inputs

The function computed by the workflow is not stable!
Small input perturbation → possibly large impact



Refresh heuristics:

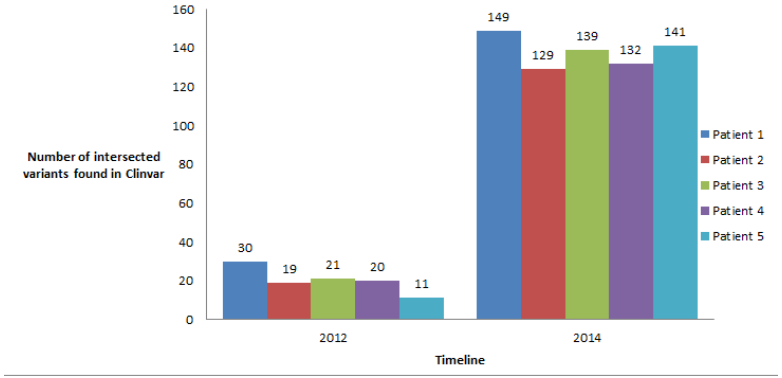
1. Changes in patient's phenotype → Refresh

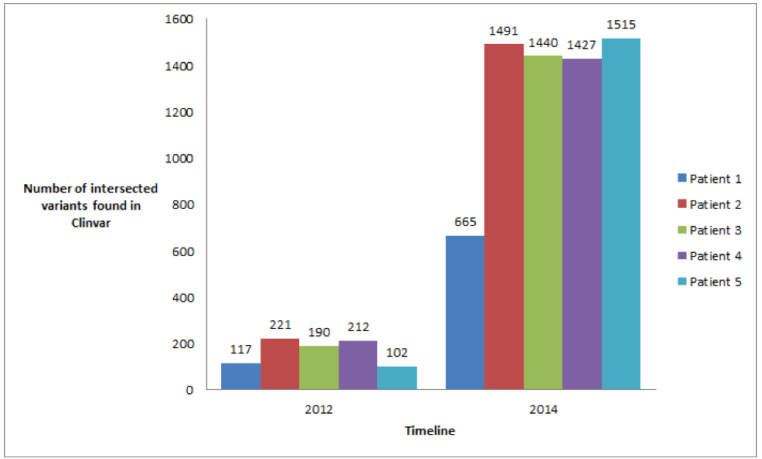
2. Changes in patient's variants:
- New variants appear → run SVI *on those new variants only*
- Variants removed (rare) → remove those from previous SVI output

Changes in data dependencies

<top-k annotated variants> = f(varset, phenotype, ClinVar, GeneMap)

ClinVar
- Monthly updates





Evolution in number of variants that affect patients
(a) with a specific phenotype
(b) Across all phenotypes

Changes in variants: ViruSurf

Frequence of single mutations in sequences collected from the population, taken at different time points

	ViruSurf	ViruSurf-GISAID	ViruSurf	ViruSurf-GISAID
	≤ 31/03/2020		≥ 01/04/2020	
With D614G	6,592	15034	23,649	18,421
Without D614G	4,664	8821	3,331	3369
D614%	58.56%	63.02%	87.65%	84.54%
Total @ **AUGUST**	61.59%		86.26%	
Total @ **OCTOBER**	42.38%		90.00%	

Frequence of variants with **high transmissibility**

Numbers change because:

i) new knowledge arises from literature,

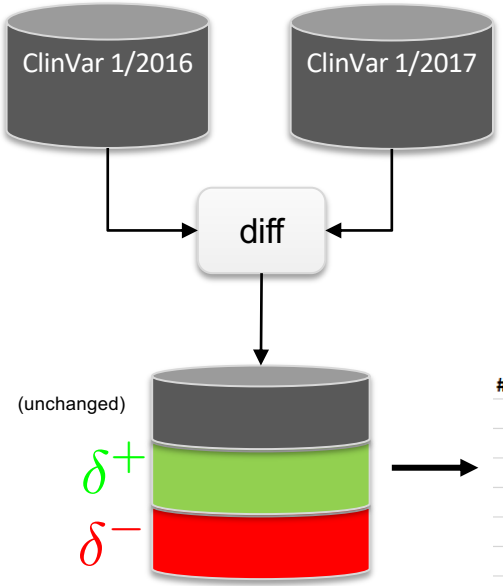
ii) variants are re-computed by eliminating low quality sequences/sub-sequences,

iii) more sequences are submitted from laboratories…

Bernasconi A., Canakoglu A., Pinoli P., Ceri S. **Empowering Virus Sequence Research through Conceptual Modeling.** In *Proceedings of the 39th International Conference on Conceptual Modeling* (ER 2020).

Canakoglu A., Pinoli P., Bernasconi A., Alfonsi T., Melidis D.P., Ceri S. **ViruSurf: an integrated database to investigate viral sequences.** *Nucleic Acids Research*, 2020. https://doi.org/10.1093/nar/gkaa846

Diff functions for SVI: ClinVar



Relational data → simple set difference

#AlleleID	Type	Name	GeneSymbol	ClinicalSignificance
15041	indel	NM_014855.2(AP5Z1):c.80_83delGGATins	AP5Z1	Pathogenic
15042	deletion	NM_014855.2(AP5Z1):c.1413_1426delGG	AP5Z1	Pathogenic
15043	single nucleotide variant	NM_014630.2(ZNF592):c.3136G>A (p.Gly1	ZNF592	Uncertain significance
15043	single nucleotide variant	NM_014630.2(ZNF592):c.3136G>A (p.Gly1	ZNF592	Uncertain significance
15071	single nucleotide variant	HOGA1, IVS, G-T, +4	HOGA1	Pathogenic
15091	deletion	XPNPEP3, 4-BP DEL, 931AACA	XPNPEP3	Pathogenic
15118	single nucleotide variant	LIPA, 934G-A	LIPA	Pathogenic
15821	single nucleotide variant	ISCU, IVS5, G-C, +382	ISCU	Pathogenic
16037	single nucleotide variant	XPA, IVS1DS, T-G, +2	XPA	Pathogenic
15091	deletion	XPNPEP3, 4-BP DEL, 931AACA	XPNPEP3	Pathogenic
15118	single nucleotide variant	LIPA, 934G-A	LIPA	Pathogenic
15120	single nucleotide variant	LIPA, IVS8, G-A, +1	LIPA	Pathogenic
15249	deletion	NM_000285.3(PEPD):c.691_693delTAC (p.	PEPD	Pathogenic
15048	single nucleotide variant	NM_000410.3(HFE):c.845G>A (p.Cys282Ty	HFE	Benign;Pathogenic;association;**not prov**
15119	duplication	NM_000235.3(LIPA):c.594dupT (p.Ala199(	LIPA	**Likely Pathogenic;** Pathogenic

The ClinVar dataset: **30** columns

Changes:
Records: **349,074** → **543,841**
Added **200,746** Removed **5,979**. Updated **27,662**

Reacting to changes in data dependencies

Impact: “Any variant with status moving from/to Red causes High impact on any patient *who is affected by the variant*”

Observation: Variants *v* within output set *y* that are in scope for patient X remain in scope! (monotonicity)

1. Variant *v* changes status
- unknown → benign
- unknown → deleterious

→ If in scope:
compare status before / after *inexpensive*

2. Brand new variant

→ recompute SVI on all inputs *expensive*

Scope: which cases are affected?

*“a change in variant **v** can only have impact on a case **X** if V and X share the same phenotype”*

SVI is unstable

Evolving knowledge about gene variations

Phenotype hypothesis	Frontotemporal Dementia-Amyotrophic Lateral Sclerosis												CADASIL	Alzheimer's disease																			
Variant file / ClinVar version	D_1071	D_1049	D_1041	D_0899	D_0854	D_0830	C_0171	C_0098	C_0056	C_0053	C_0051	B_0307	D_1136	C_1457	C_0072	C_0071	C_0068	C_0065	B_0396	B_0384	B_0370	B_0365	B_0358	B_0338	B_0331	B_0229	B_0214	B_0209	B_0208	B_0203	B_0202	B_0201	B_0179
08/15	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
09/15	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
10/15	·	·	·	·	·	·	·	·	·	·	·	·	■	□	·	·	·	□	·	·	·	□	·	·	□	□	·	□	□	·	·	□	□
11/15	□	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
12/15	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
01/16	·	·	·	·	·	·	·	·	·	·	·	·	·	□	□	·	□	□	□	·	□	□	□	□	□	□	□	□	□	·	□	□
02/16	·	·	□	·	□	·	□	·	□	□	□	□	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
03/16	□	·	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
04/16	·	·	·	·	·	·	·	·	·	·	·	·	·	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
05/16	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	□	·
06/16	·	·	□	·	□	·	□	□	·	·	□	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
07/16	·	·	□	·	□	·	·	·	□	·	□	·	·	·	□	·	·	·	·	□	□	·	·	□	·	□	□	·	□	□	□	□	□
08/16	·	·	·	·	·	·	·	·	·	·	·	□	·	❖	■	❖	□	■	□	·	■	■	■	■	□	■	■	□	□	■	·	□	□
09/16	·	·	·	·	·	·	·	·	·	·	·	·	·	·	□	·	□	□	□	·	□	□	□	□	□	□	□	□	·	□	·	·	□
10/16	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

Should we care about updates?

Sparsity issue:

- About 500 executions
- 33 patients
- total runtime about 60 hours
- Only 14 relevant output changes detected

4.2 hours of computation per change

Empirical evaluation

Phenotype hypothesis		Frontotemporal Dementia-Amyotrophic Lateral Sclerosis												CADASIL	Alzheimer's disease																		CMS	
GeneMap version	ClinVar version	B_0307	C_0051	C_0053	C_0056	C_0098	C_0171	D_0830	D_0854	D_0899	D_1041	D_1049	D_1071	D_1136	B_0198	B_0201	B_0202	B_0203	B_0208	B_0209	B_0214	B_0229	B_0331	B_0338	B_0358	B_0365	B_0370	B_0384	B_0396	C_0065	C_0068	C_0071	C_0072	C_1457	M_0785	M_0789
161101	1611	·	□	·	·	□	·	·	·	·	·	·	□	·	□	·	·	·	□	□	·	□	□	·	·	□	·	·	·	□	·	·	·	□	□	□
161201	1612	·	□	·	·	□	·	·	·	·	·	·	·	■	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	□	□
170101	1701	□	□	□	□	□	□	□	□	□	□	□	□	□	■	□	□	❖	□	■	❖	❖	■	❖	❖	❖	❖	□	❖	❖	❖	□	❖	□	□	□
170201	1702	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
170302	1703	·	·	·	·	·	·	·	·	·	·	·	·	·	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
170401	1704	□	□	□	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	■	■	■	□	□
170501	1705	□	□	□	□	□	□	□	□	□	□	□	□	·	□	□	□	□	□	·	□	□	□	□	□	□	□	□	□	□	□	·	□	□	□	□
170601	1706	·	·	·	□	·	·	·	·	·	·	·	·	□	□	·	·	·	□	□	·	□	□	·	·	□	·	·	·	□	·	·	·	□	□	□
170701	1707	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
170801	1708	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	·	□	□	□	□	□	□	□	□	□	□	·	□	□	□	□	□	□	□	□
170901	1709	·	·	·	·	·	·	·	·	·	·	·	·	·	□	□	·	□	·	□	□	□	□	□	□	□	□	·	□	·	·	·	·	·	□	□	□
171001	1710	□	□	□	□	□	□	□	□	□	□	□	□	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	□	□

re-executions 495 → 71 Ideal: 14

But: no false negatives

General Problem formulation

$f : X \times D \rightarrow Y$ $\quad \mathbf{y} = f(\mathbf{x}, \mathbf{d})$ d: current state of data source D

Changes in *d:* $\mathbf{d} \rightsquigarrow \mathbf{d}'$ $\quad \delta_D(\mathbf{d}, \mathbf{d}')$ $\quad \mathbf{y} = f(\mathbf{x}, \mathbf{d}')$ $\quad \mathbf{y} = f(\mathbf{x}', \mathbf{d}')$

$\mathbf{x} \rightsquigarrow \mathbf{x}'$ $\quad \delta_X(\mathbf{x}, \mathbf{x}')$ $\quad \mathbf{y} = f(\mathbf{x}', \mathbf{d})$

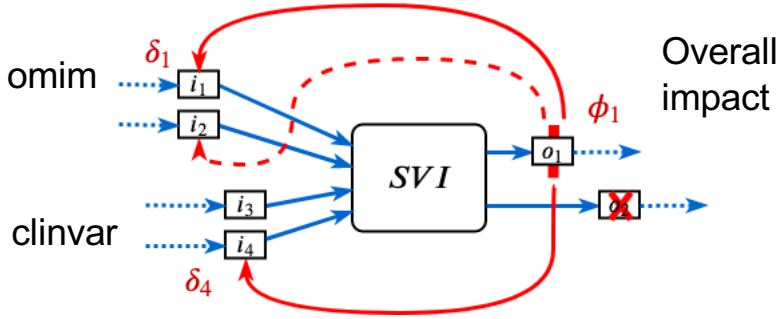
Impact due to changes in d: $imp_D(\mathbf{d}, \mathbf{d}', \mathbf{x}) = \delta_Y(f(\mathbf{x}, \mathbf{d}), f(\mathbf{x}, \mathbf{d}'))$

Problem: for each ***x*** in *X*
estimate impact without computing ***f(x,d)***
(assuming f is not stable)

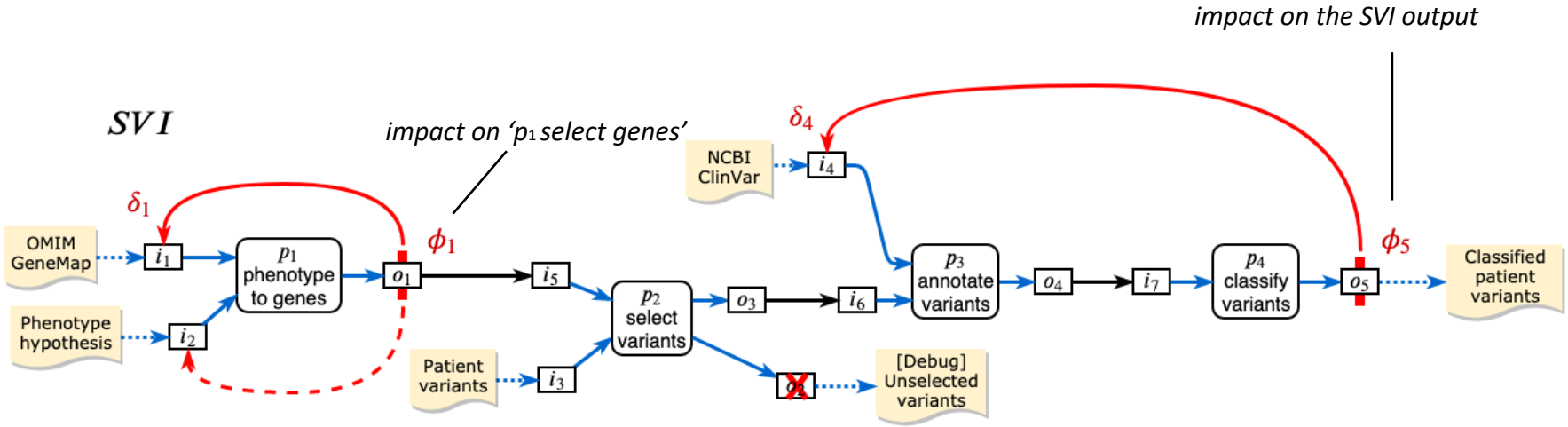
Impact estimation: $\widehat{imp}_D(\mathbf{d}, \mathbf{d}', \mathbf{x}) = ReComp(\delta_D(\mathbf{d}, \mathbf{d}', \mathbf{x}), f(x)) \in \{0, 1\}$

Scope of impact: $\{\mathbf{x} \in X | \widehat{imp}_D(\mathbf{d}, \mathbf{d}', \mathbf{x}) = 1\}$

Diff and impact functions for white box process



- Data-specific
- Process-specific

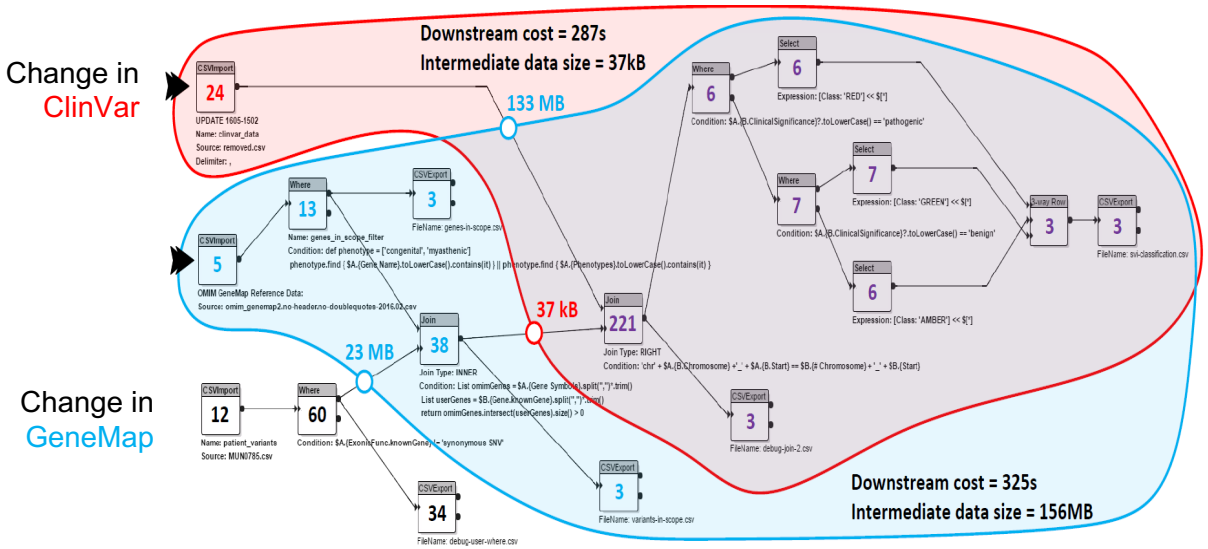


White box ReComp and Process Provenance

White box process structure →

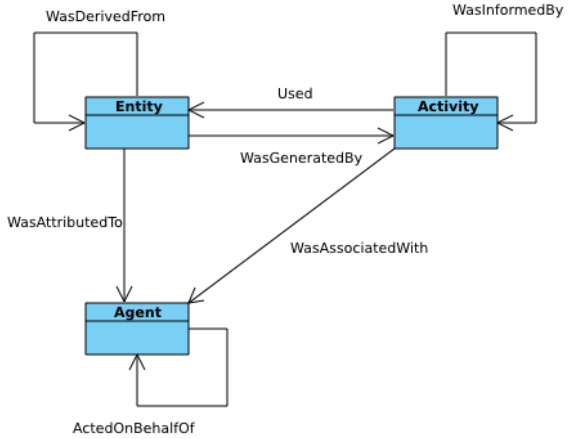
detailed history of each execution →

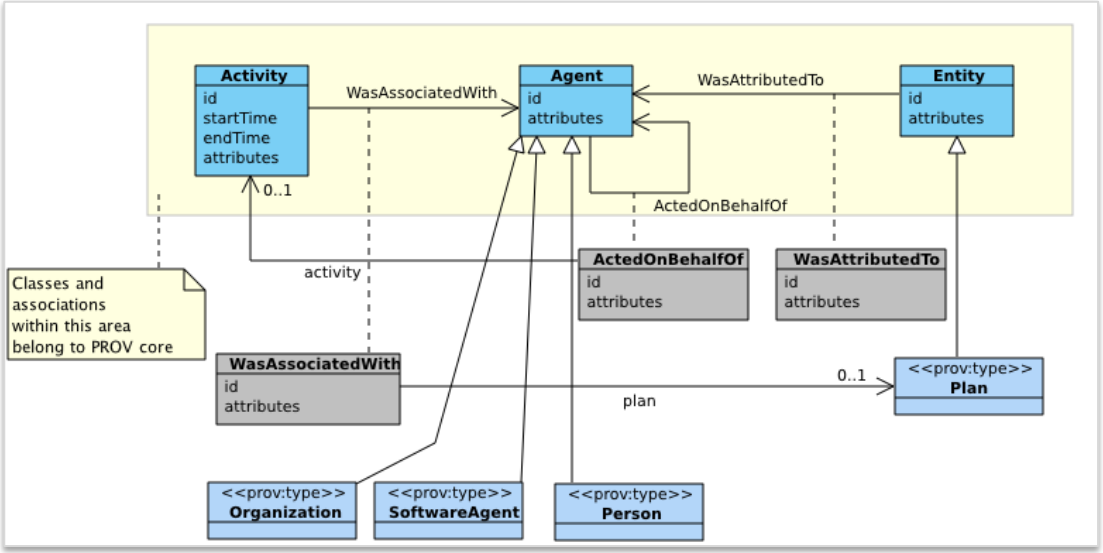
more granular optimisations become possible



PROV-DM: The PROV Data Model

W3C Recommendation 30 April 2013



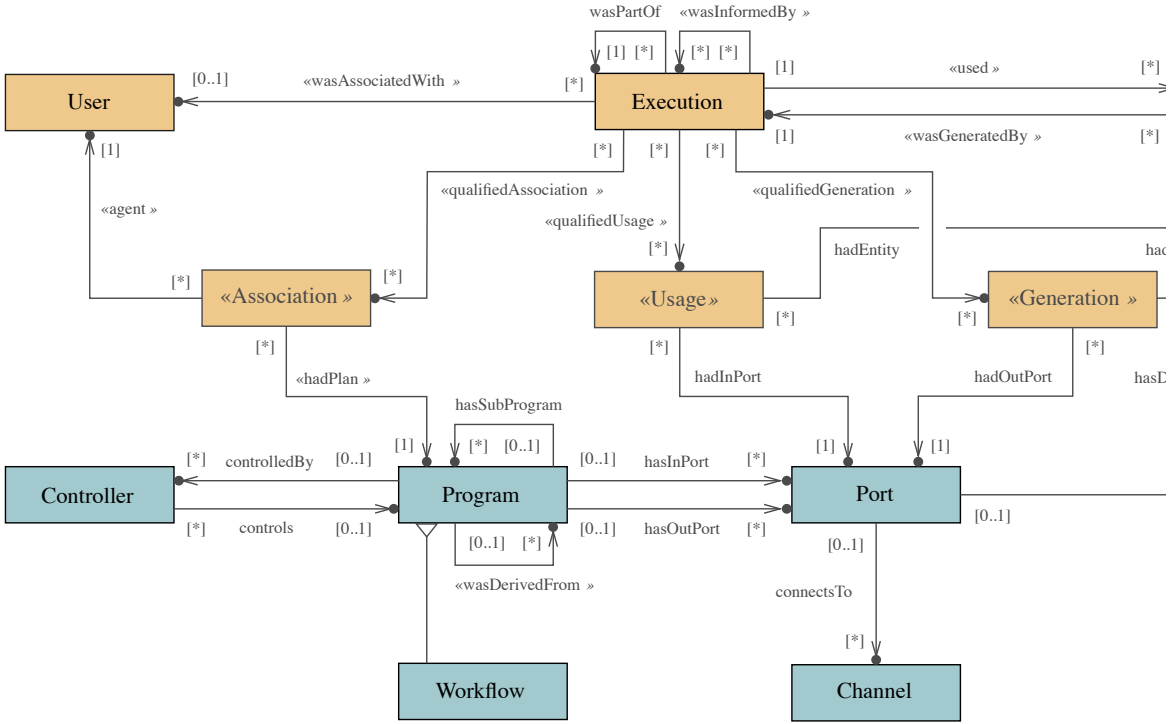


https://www.w3.org/TR/prov-dm/

Prov + process structure = ProvONE

ProvONE: A PROV Extension Data Model for Scientific Workflow Provenance

Draft 01 May 2016

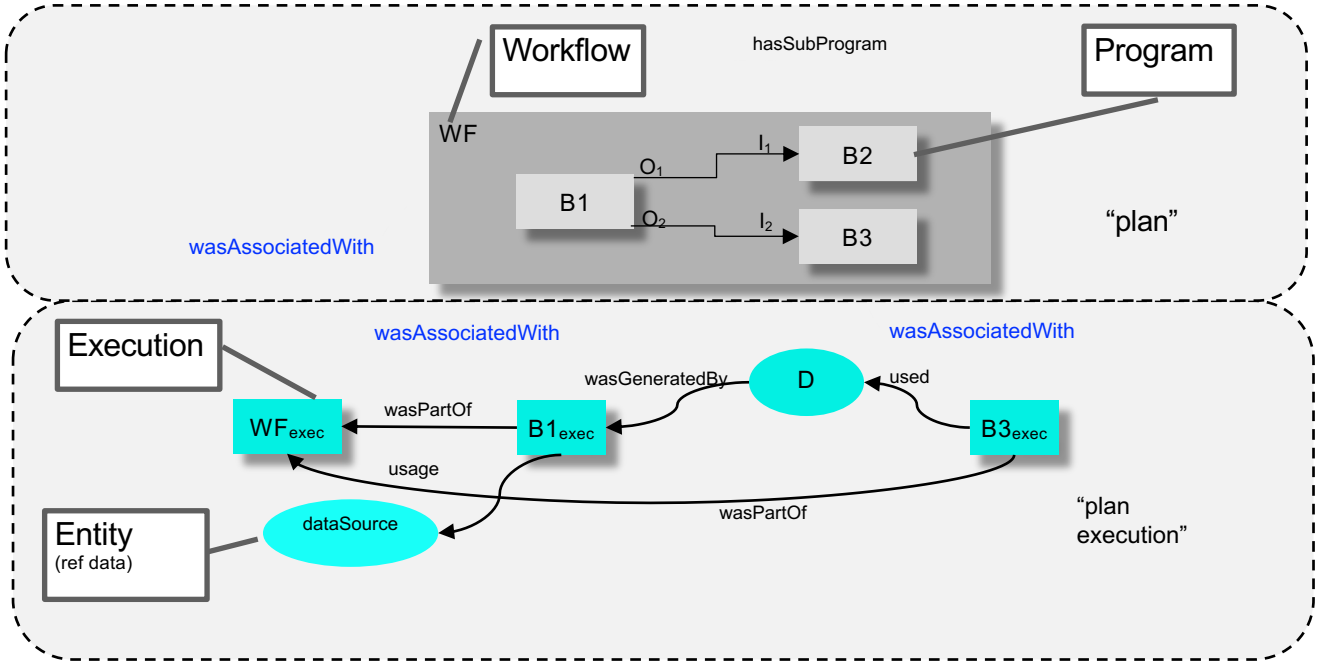


Workflow Provenance: example

ProvONE combines process structure with runtime provenance

```
wf isa Workflow
hasSubProgram(wf, B1)
hasSubProgram(wf, B2)
hasSubProgram(wf, B3)
hasOutPort(B3,O1)
hasInPort(B3,I2)
connectsTo(O2, c)
connectsTo(I2, c)
…
```

```
wasAssociatedWith(wf_exec, wf)
wasAssociatedWith(B1_exec, B1)
wasAssociatedWith(B3_exec, B3)
wasPartOf(B1_exec, wf_exec)
wasPartOf(B3_exec, wf_exec)
wasGeneratedBy(g,D,B1_exec)
used(u,B3exec,D)
hadInPort(u, I2)
hadOutPort(g, O2)
```



$B1_{exec}$ is an instance of program B1, $B3_{exec}$ is an instance of program B3, which are part of Workflow wf

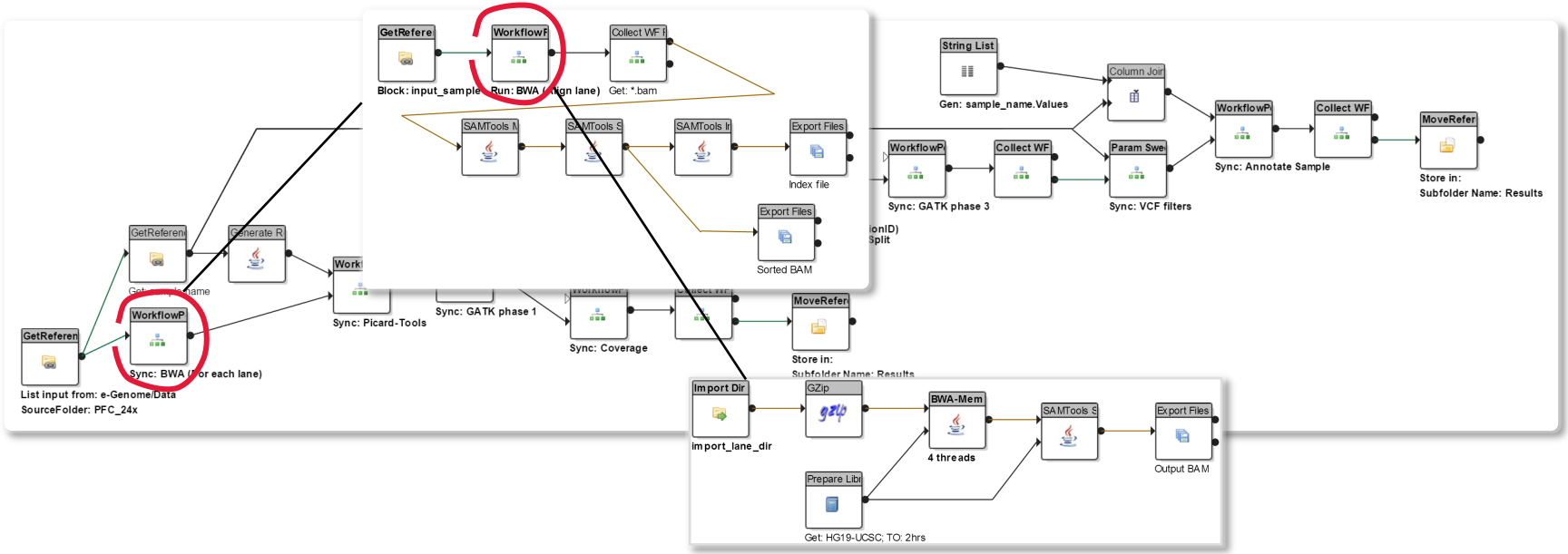
B3 has input port I_2; B1 has output port O_2, and these are connected (through channel c)

Data item D was generated by $B1_{exec}$ on port O_2 and used by $B3_{exec}$ on port I_2

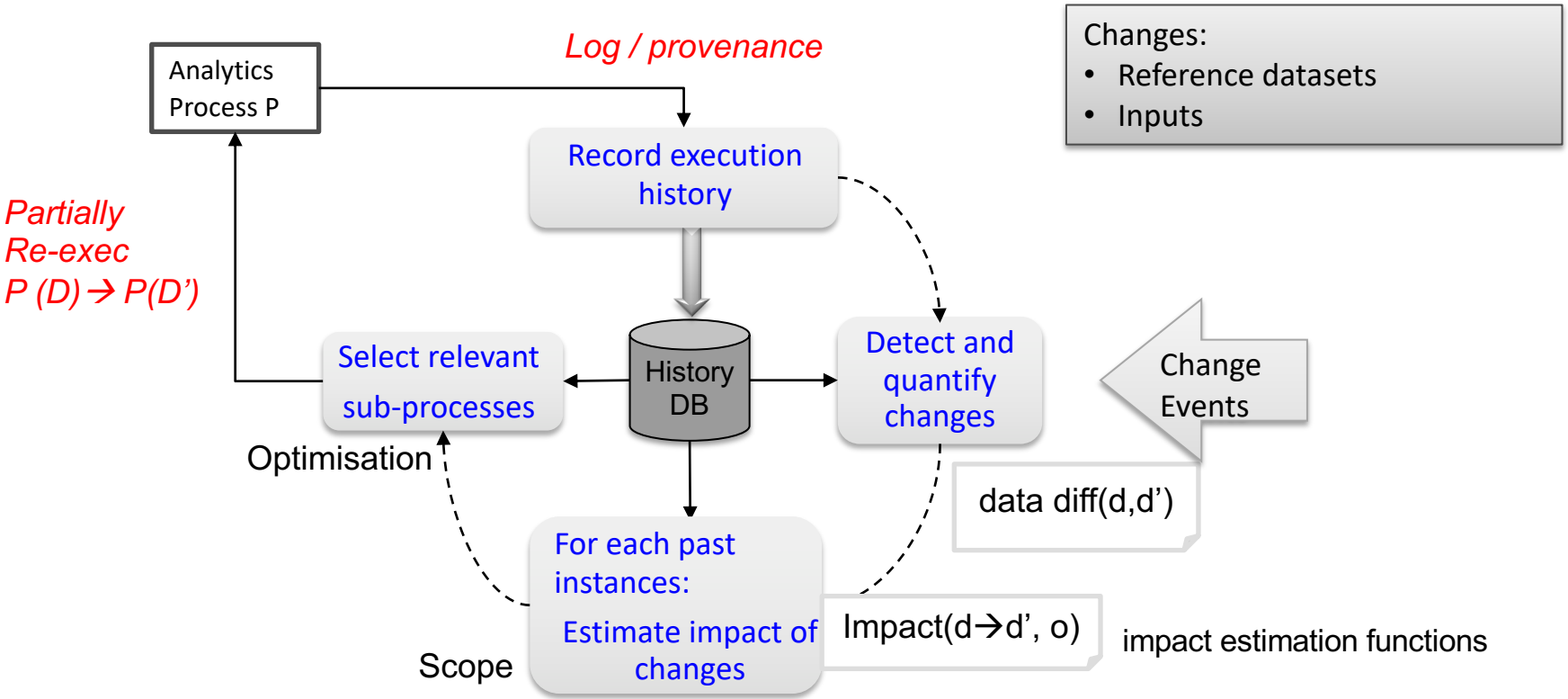
ProvONE enables selective re-execution

ProvONE traces the execution of nested workflows

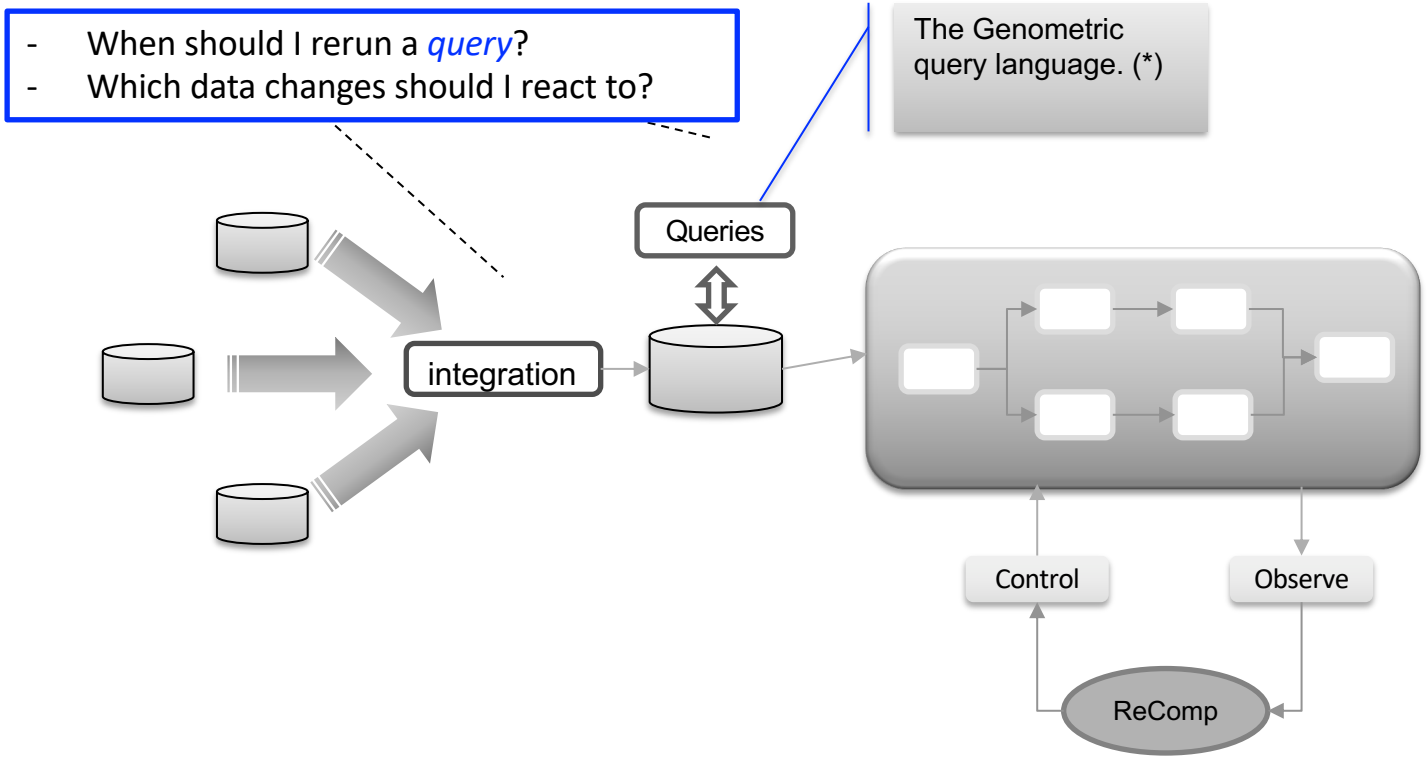
Enables selective re-computation of the fragments affected by a change



Summary: the ReComp meta-process



End-to-end ReComp



(*) Masseroli M, Canakoglu A, Pinoli P, Kaitoua A, Gulino A, Horlova O, Nanni L, Bernasconi A, Perna S, Stamoulakatou E, Ceri S. **Processing of big heterogeneous genomic datasets for tertiary analysis of Next Generation Sequencing data.** *Bioinformatics*, 2018. https://doi.org/10.1093/bioinformatics/bty688

The GenoMetric Query Language

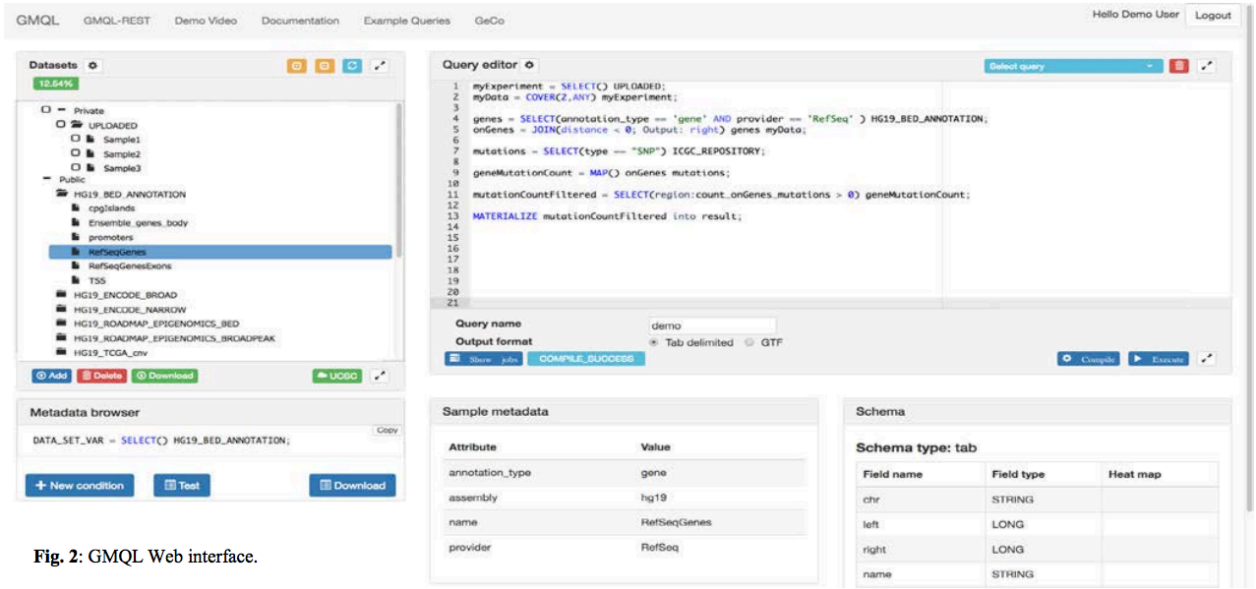


Fig. 2: GMQL Web interface.

A query processing environment for supporting, at a high level of abstraction, data extraction and the most common data-driven computations required by tertiary data analysis of Next Generation Sequencing datasets

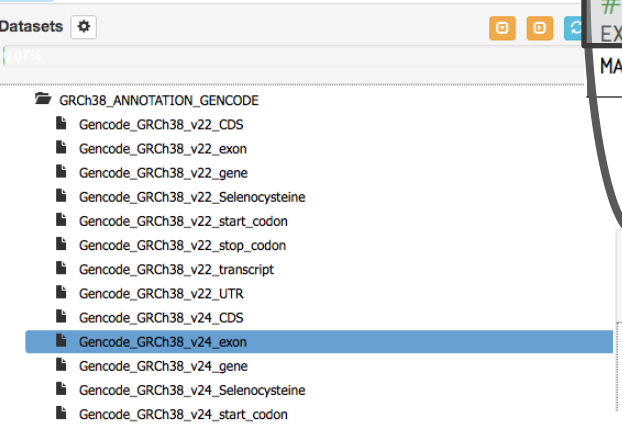
Masseroli M, Canakoglu A, Pinoli P, Kaitoua A, Gulino A, Horlova O, Nanni L, Bernasconi A, Perna S, Stamoulakatou E, Ceri S. **Processing of big heterogeneous genomic datasets for tertiary analysis of Next Generation Sequencing data.** *Bioinformatics*, 2018. https://doi.org/10.1093/bioinformatics/bty688

ReComp for GenoMetric Query Language 1

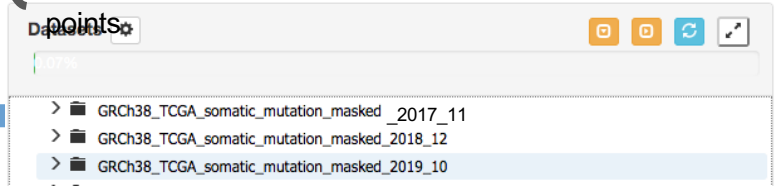
For kidney cancers, find mutations and their number in each exon

```
#Select mutation data based on both region and metadata attributes
MUT = SELECT(gdc__primary_site == "Kidney" AND gdc__disease_type == "Adenomas and Adenocarcinomas" AND
clinical__shared__history_of_neoadjuvant_treatment == "No" AND
clinical__clin_shared__followup_treatment_success == "Complete Remission/Response";
region: dbsnp_rs == "novel" GRCh38_TCGA_somatic_mutation_masked_2019_10;
#Select known human protein-coding and non-protein-coding exon regions of the GENCODE annotation release 27
EXON = SELECT(annotation_type == "exon" AND release_version == "27") GRCh38_ANNOTATION_GENCODE;
#Map the mutations to the exons and count how many they are in each exon of each sample
EXON_MUT = MAP(count_name: MUT_count) EXON MUT;
#Remove exons that do not contain mutations
EXON_MUT_SELECT = SELECT(region: MUT_count > 0) EXON_MUT;
#In the metadata of each sample add the count of how many exons remain and the maximum number of mutations in
#an exon of the sample
EXON_RES = EXTEND(exon_count AS COUNT(), max_mut AS MAX(MUT_count)) EXON_MUT_SELECT;
MATERIALIZE EXON_RES INTO result1_exons_mutations;
```

Different annotation files are released periodically



Different somatic mutation datasets at different time points



Recomputation of the query may only be required if we can estimate that > 10% samples would obtain a changed count

ReComp for GenoMetric Query Language 3

Find distal bindings in transcription regulatory regions: Find all enriched regions (peaks) of CTCF transcription factor (TF) in ENCODE ChIP-seq narrow peak samples from GM12878 lymphoblastoid human cell line which are the nearest regions farther than 100 kb from a transcription start site (TSS). For the same cell line, find also all peaks of the H3K4me1 histone modification (HM) which are also the nearest regions farther than 100 kb from a TSS. Then, out of the TF and HM peaks found in the same cell line, return all TF peaks that overlap with at least a HM peak and known enhancer (EN) region.

```
TF = SELECT(assay == "ChIP-seq" AND output_type == "peaks" AND
        experiment_target == "CTCF-human" AND
        biosample_term_name == "GM12878") HG19_ENCODE_NARROW_NOV_2017
HM = SELECT(assay == "ChIP-seq" AND output_type == "peaks" AND
        experiment_target == "H3K4me1-human" AND
        biosample_term_name == "GM12878") HG19_ENCODE_NARROW_NOV_2017
TSS = SELECT(annotation_type == "TSS" AND provider == "UCSC") HG19_BED_ANNOTATION
EN = SELECT(annotation_type == "enhancer" AND provider == "UCSC")
        HG19_BED_ANNOTATION

TF1 = JOIN(DISTANCE > 100000, MINDISTANCE(1); output: RIGHT_DISTINCT) TSS TF;
HM1 = JOIN(DISTANCE > 100000, MINDISTANCE(1); output: RIGHT_DISTINCT) TSS HM;
HM2 = JOIN(DISTANCE < 0; output: INT) EN HM1;
HM3 = MERGE() HM2;
TF_RES = JOIN(DISTANCE < 0; output: RIGHT_DISTINCT) HM3 TF1;

MATERIALIZE TF_RES INTO TF_RES;
```

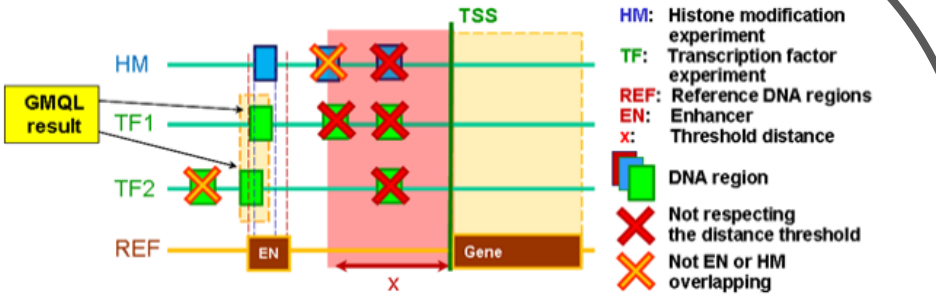
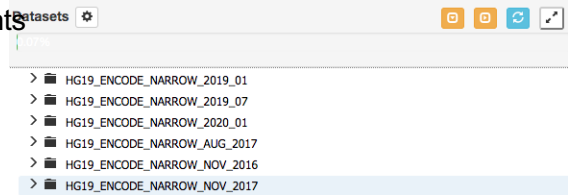
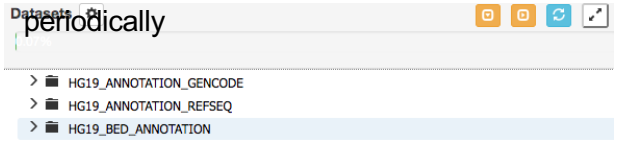


Figure 1. The histone modification (HM) and transcription factor (TF) binding site enriched regions ('peaks'), known reference DNA regions and their distance relationships involved in Example 1.

Different Transcription Factors datasets at different time points



Different annotation files are released periodically



Recomputation of the query may only be required if we can estimate a considerable change

<event name>

Summary and selected references

ReComp is a customizable framework to enable optimisations of data-intensive processes

It reacts to changes in data by trying to predict their impact on process outcomes

When provenance is available, it can be used to optimize process re-run

Prototype implementation exists

Extensions:
- controlling refresh of query results
- queries on data source establish which data changes we care about

- P. Missier and J. Cala, “Efficient Re-computation of Big Data Analytics Processes in the Presence of Changes: Computational Framework, Reference Architecture, and Applications,” in *Procs. IEEE Big Data Congress*, 2019.
- J. Cała and P. Missier, “Selective and Recurring Re-computation of Big Data Analytics Tasks: Insights from a Genomics Case Study,” *Big Data Res.*, vol. 13, pp. 76–94, Sep. 2018.
- J. Cala and P. Missier, “Provenance Annotation and Analysis to Support Process Re-Computation,” in *Procs. IPAW 2018*, 2018.