Phenoflow

Clinical Natural Language Processing Group, University of Edinburgh

Martin Chapman, Research Fellow in Phenomics (GSTT BRC and HDR UK)

King's College London

Overview

Background

Definition: EHR-based phenotype definition i

An electronic health record (EHR)-based phenotype definition is an **abstract specification** that details how to extract a **cohort** of patients from a set of health records who all exhibit the **same disease or condition**.

Definition: EHR-based phenotype definition ii

Table 1: Phenotype definition formats

Format	Description	Example	Category
Code list	A set of codes that must exist in a patient's health record in order to include them within a phenotype cohort	COVID-19 ICD-10 code *U07.1*	Rule-based
Simple data elements	Formalising the relationship between code-based *data elements* using *logical connectives*	COVID-19 ICD-10 code *U07.1* AND ICD-11 code *RA01.0*	Rule-based
Complex data elements	Formalising the relationship between complex data elements, such as those derived via NLP.	Patient's blood pressure reading $>$ 140 OR patient notes contain 'high BP'	Rule-based
Temporal	Prefix rules with temporal qualifiers	Albumin levels increased by 25% over 6 hours, high blood pressure reading has to occur during hospitalisation.	Rule-based
Trained classifier	Use rule-based definitions as the basis for constructing a classifier for future (or additional) cohorts	A k-fold cross validated classifier capable of identifying COVID-19 patients	Probabilistic

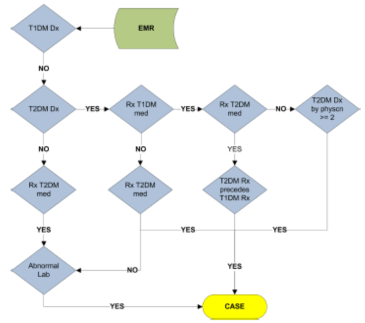
Definition: Computable phenotype i

Each definition is realised as one or more **computable phenotypes** for a given dataset (e.g. an SQL script, Python code, etc.).

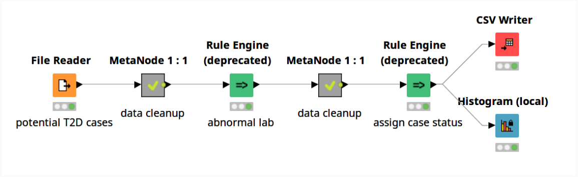
Definition: Computable phenotype ii

A Type 2 Diabetes (T2DM) phenotype:

Figure 1: Algorithm for identifying T2DM cases in the EMR.



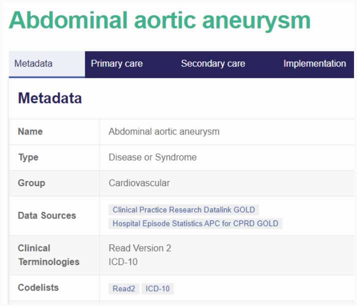
Definition



```
SELECT UserID, COUNT(DISTINCT AbnormalLab) AS abnormal_lab
FROM Patients
GROUP BY UserID
HAVING abnormal_lab > 0;
...
```

Computable forms

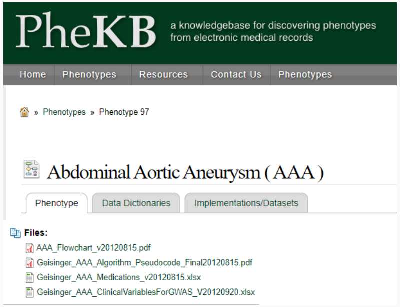
Phenotype definition landscape i


Abdominal aortic aneurysm

Metadata

Name	Abdominal aortic aneurysm
Type	Disease or Syndrome
Group	Cardiovascular
Data Sources	Clinical Practice Research Datalink GOLD Hospital Episode Statistics APC for CPRD GOLD
Clinical Terminologies	Read Version 2 ICD-10
Codelists	Read2 ICD-10

portal.caliberresearch.org


PheKB
a knowledgebase for discovering phenotypes from electronic medical records
Home
Phenotypes
Resources
Contact Us
Phenotypes
» Phenotypes » Phenotype 97

Abdominal Aortic Aneurysm (AAA)

Phenotype
Data Dictionaries
Implementations/Datasets

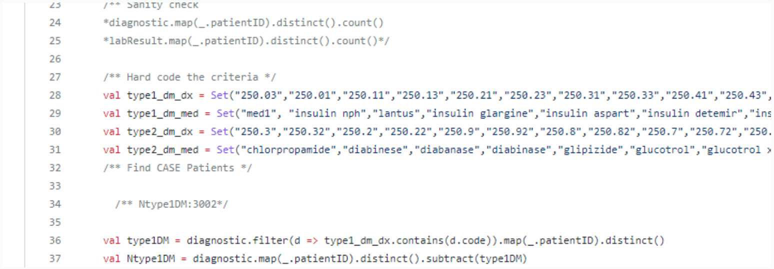
Files:
- AAA_Flowchart_v20120815.pdf
- Geisinger_AAA_Algorithm_Pseudocode_Final20120815.pdf
- Geisinger_AAA_Medications_v20120815.xlsx
- Geisinger_AAA_ClinicalVariablesForGWAS_V20120920.xlsx

phekb.org

Computable form often **omitted** – this makes it unclear how to **implementation and execute a definition in practice** against a dataset, particularly for **non-technical users**.

(We'll revisit CALIBER's **implementation** tab shortly.)

Phenotype definition landscape ii



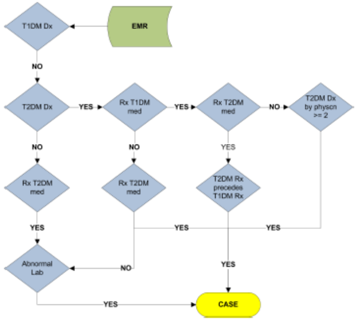
```
/** Sanity check
*diagnostic.map(_.patientID).distinct().count()
*labResult.map(_.patientID).distinct().count()*/

/** Hard code the criteria */
val type1_dm_dx = Set("250.03","250.01","250.11","250.13","250.21","250.23","250.31","250.33","250.41","250.43",
val type1_dm_med = Set("med1", "insulin nph","lantus","insulin glargine","insulin aspart","insulin detemir","ins
val type2_dm_dx = Set("250.3","250.32","250.2","250.22","250.9","250.92","250.8","250.82","250.7","250.72","250.
val type2_dm_med = Set("chlorpropamide","diabinese","diabanase","diabinase","glipizide","glucotrol","glucotrol x
/** Find CASE Patients */

  /** Ntype1DM:3002*/

val type1DM = diagnostic.filter(d => type1_dm_dx.contains(d.code)).map(_.patientID).distinct()
val Ntype1DM = diagnostic.map(_.patientID).distinct().subtract(type1DM)
```

Conversely, if included, the definition and computable form are often conflated as a **single executable** - an R script on Github is **not suitably abstract** to be a phenotype definition itself.

Chapman, Martin, et al. "Desiderata for the development of next-generation electronic health record phenotype libraries." GigaScience, 2021.

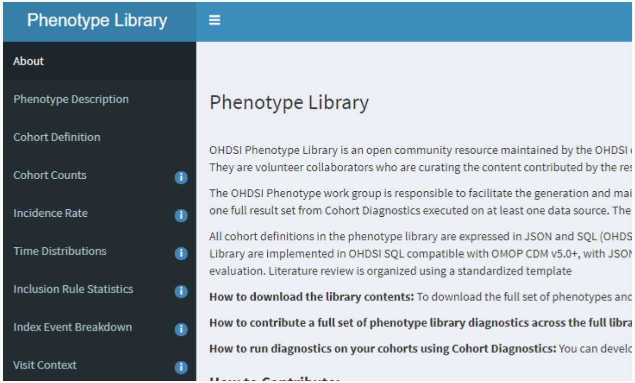
Phenotype definition landscape iii

Figure 1: Algorithm for identifying T2DM cases in the EMR.



Otherwise simple definitions are often made **complex** by **idiosyncratic terminology** and a **convoluted structure**.

Phenotype definition landscape iv



Phenotype Library

OHDSI Phenotype Library is an open community resource maintained by the OHDSI
They are volunteer collaborators who are curating the content contributed by the re

The OHDSI Phenotype work group is responsible to facilitate the generation and ma
one full result set from Cohort Diagnostics executed on at least one data source. The

All cohort definitions in the phenotype library are expressed in JSON and SQL (OHDS
Library are implemented in OHDSI SQL compatible with OMOP CDM v5.0+, with JSO
evaluation. Literature review is organized using a standardized template

How to download the library contents: To download the full set of phenotypes an

How to contribute a full set of phenotype library diagnostics across the full libr

How to run diagnostics on your cohorts using Cohort Diagnostics: You can devel

https://data.ohdsi.org/PhenotypeLibrary

Tied to a **single standard**, e.g. OHDSI's gold standard phenotype library and the OMOP CDM.

Why are these things a problem?

We want to be able to **reuse** definitions as much as possible, to enable cohorts of patients with a given condition to be identified as **efficiently** and **consistently** as possible, **within the same domain** (e.g. research, clinical trials, decision-support).

- We are *not* looking for a single, **canonical** version of each definition across domains – it is perfectly possible for there to be **multiple definitions** for the same condition depending on use case.

Why are these things a problem?

We want to be able to **reuse** definitions as much as possible, to enable cohorts of patients with a given condition to be identified as **efficiently** and **consistently** as possible, **within the same domain** (e.g. research, clinical trials, decision-support).

- We are *not* looking for a single, **canonical** version of each definition across domains – it is perfectly possible for there to be **multiple definitions** for the same condition depending on use case.

The current landscape is not always conducive to reuse:

- The lack of a computable form, or guidance on how to derive one, reduces definition **portability** (the **ease** with which a definition can be implemented).
- A convoluted structure reduces definition **reproducibility** (the **accuracy** with which a definition can be implemented).

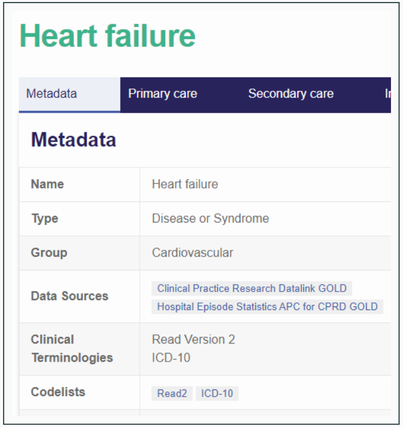
Phenotype models

Phenotype models i

Phenotype models govern the **information required** for, and the **structure** of, phenotype definitions.

- They may, for example, govern the **logical connectives** available to a definition author when producing a definition (e.g. conjunction and disjunction).
- Many definitions have an **inherent** model, such as the fields that are required when the definitions is stored in a **phenotype library**.
- Models may also be derived from existing (non-executable) **modelling languages**, such as the Clinical Quality Language (CQL).

Phenotype models ii

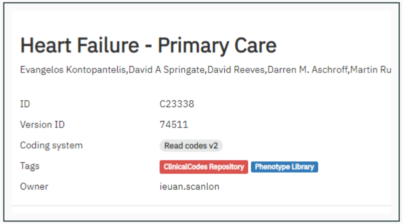


Heart failure

Metadata | Primary care | Secondary care | In

Metadata

Name	Heart failure
Type	Disease or Syndrome
Group	Cardiovascular
Data Sources	Clinical Practice Research Datalink GOLD Hospital Episode Statistics APC for CPRD GOLD
Clinical Terminologies	Read Version 2 ICD-10
Codelists	Read2 ICD-10



Heart Failure - Primary Care

Evangelos Kontopantelis,David A Springate,David Reeves,Darren M. Aschroff,Martin Ru

ID	C23338
Version ID	74511
Coding system	Read codes v2
Tags	ClinicalCodes Repository Phenotype Library
Owner	ieuan.scanlon

```
library "PhEMA Heart Failure" version '1.0.0'

using QUICK

codesystem "ActCodes": 'http://hl7.org/fhir/v3/ActCode'

valueset "Echo VS": '2.16.840...'
valueset "HF Dx VS": '2.16.840...'

code "Inpatient Encounter": 'IMP' from "ActCodes"
code "Outpatient Encounter": 'AMB' from "ActCodes"
```

Phenotype model requirements

While a **standard structure** goes some way towards improving definition clarity, the use of an **explicit** phenotype model can help **address many of the issues** we've seen, but to be effective, a model must fulfil certain **requirements**:

Phenotype model requirements

While a **standard structure** goes some way towards improving definition clarity, the use of an **explicit** phenotype model can help **address many of the issues** we've seen, but to be effective, a model must fulfil certain **requirements**:

1. Needs to **connect**, yet keep **distinct**, a phenotype definition and its computable form.
   1.1 A definition needs to remain suitably **abstract** while making provision for an associated **computable counterpart**. Ideally facilitate **one-to-many connectivity**, connecting with multiple implementations of the same logic.

Phenotype model requirements

While a **standard structure** goes some way towards improving definition clarity, the use of an **explicit** phenotype model can help **address many of the issues** we've seen, but to be effective, a model must fulfil certain **requirements**:

1. Needs to **connect**, yet keep **distinct**, a phenotype definition and its computable form.
   1.1 A definition needs to remain suitably **abstract** while making provision for an associated **computable counterpart**. Ideally facilitate **one-to-many connectivity**, connecting with multiple implementations of the same logic.
2. Needs to prioritise **clarity** to combat the complexity of definitions.

Phenotype model requirements

While a **standard structure** goes some way towards improving definition clarity, the use of an **explicit** phenotype model can help **address many of the issues** we've seen, but to be effective, a model must fulfil certain **requirements**:

1. Needs to **connect**, yet keep **distinct**, a phenotype definition and its computable form.
    1.1 A definition needs to remain suitably **abstract** while making provision for an associated **computable counterpart**. Ideally facilitate **one-to-many connectivity**, connecting with multiple implementations of the same logic.
2. Needs to prioritise **clarity** to combat the complexity of definitions.
3. Support a variety of **target data format**s.

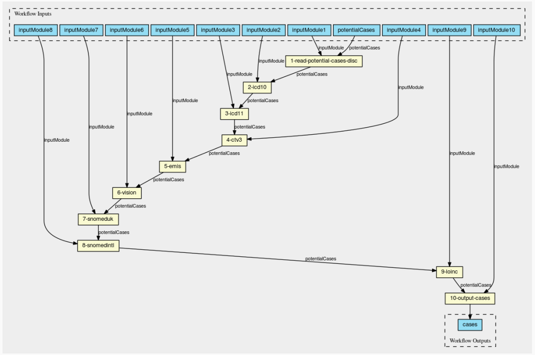
Phenotype model requirements

While a **standard structure** goes some way towards improving definition clarity, the use of an **explicit** phenotype model can help **address many of the issues** we've seen, but to be effective, a model must fulfil certain **requirements**:

1. Needs to **connect**, yet keep **distinct**, a phenotype definition and its computable form.
    1.1 A definition needs to remain suitably **abstract** while making provision for an associated **computable counterpart**. Ideally facilitate **one-to-many connectivity**, connecting with multiple implementations of the same logic.
2. Needs to prioritise **clarity** to combat the complexity of definitions.
3. Support a variety of **target data format**s.
4. Accommodate (and potentially combine) all definitions **types**.

Phenoflow workflow-based model i

Phenoflow workflow-based phenotypes are a step of sequential **steps**, which effectively transition a **population** of patients to a **cohort** that exhibit the condition captured.



Phenoflow workflow-based model ii

Each step in the model consists of three layers:

- **Abstract** Expresses the **logic** of that step. Says nothing about **implementation**.
- **Functional** Specifies the **inputs** to, and **outputs** from, this step (metadata) e.g., the format of an intermediate cohort.
- **Computational** Defines an environment for the **execution** of one or more **implementation units** (e.g. a script, data pipeline module, etc.).

Phenoflow workflow-based model iii

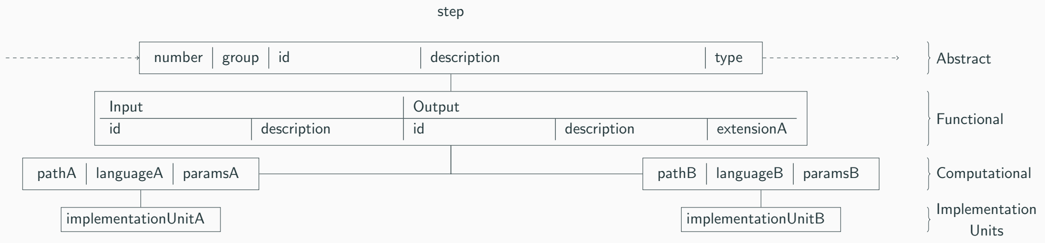
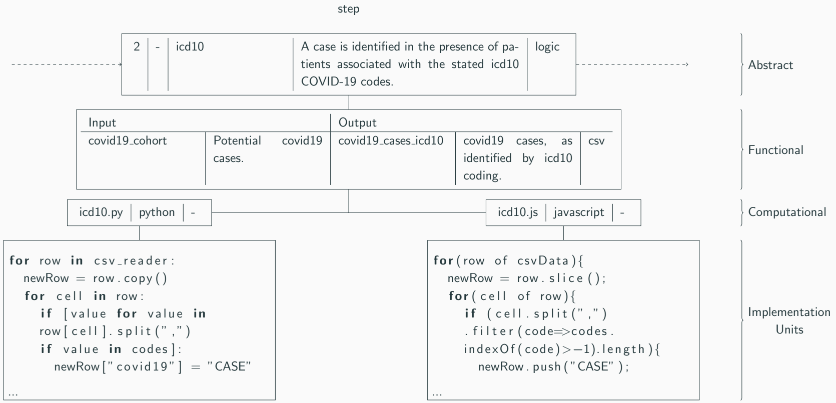


Figure 1: Structured phenotype definition model (step) and implementation units.

(1) Separate, yet connect, a phenotype definition with its computable form

The separation of the model into logic and implementation layers provides the required **connectivity** with a computable form, without affecting **abstraction**:



```
for row in csv_reader:
  newRow = row.copy()
  for cell in row:
    if [value for value in
    row[cell].split(",")
    if value in codes]:
      newRow["covid19"] = "CASE"
...
```

```
for(row of csvData){
  newRow = row.slice();
  for(cell of row){
    if (cell.split(",")
    .filter(code=>codes.
    indexOf(code)>-1).length){
      newRow.push("CASE");
...
```

Figure 2: Individual step of COVID-19 code-based Phenoflow definition and implementation units.

(2) Prioritise clarity to combat the complexity of definitions

On top of definitions now having an expected structure, separation into steps provides a **logical flow**.

Each step provides **three descriptions** of the functionality it contains, to aid clarity:

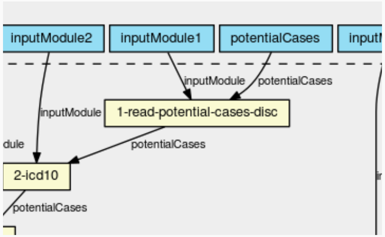
1. A single **ID**, providing an overview of the step's functionality.
2. A longer **description** of the functionality contained within the step.
3. A **classification** of the step under a pre-defined ontology, so that even if the ID and description are not sufficient, a general understanding of the functionality of the step can still be extracted[1].

Inputs and **outputs** to each step provide further information.

[1]As of now we still use simple classification, e.g. logic, but finer a granularity of types is forthcoming.

(3) Support a variety of target data formats

We add additional **contraints** to the workflow-based structure to dictate that the first step in a definition is always a **data read** (and the last is always a **cohort output**).



Because of the **modularity** of the model structure, we are able to **swap in and out** the logic, and associated implementation, of the data read step – while the other steps remain **unchanged** – in order to accommodate **multiple data formats**.

More on this **connector** approach shortly.

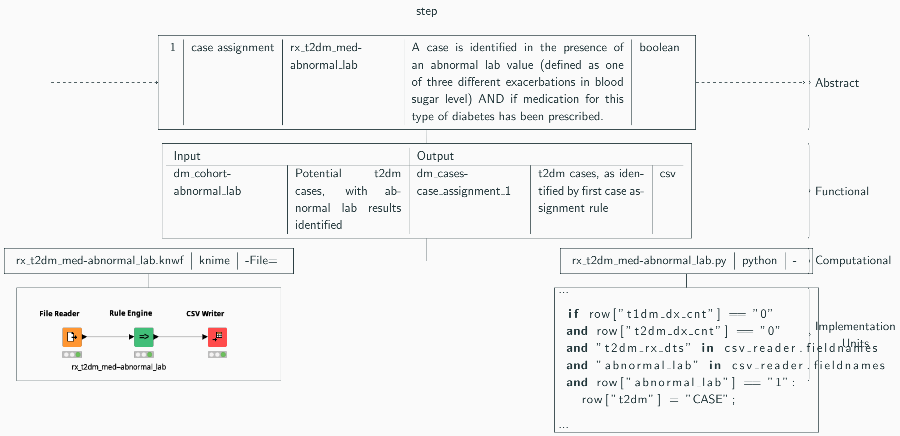
(4) Accommodate (and potentially combine) all definitions types i

The **generality** of the model allows it to capture information relating to a wide range of different definition types.

Similarly, the separation of logic into steps, with clear inputs and outputs, makes each step **self-contained**, allowing types to be mixed within a **single definition**.

- One step may identify patients based on a list of codes, while a subsequent step may describe the use of more complex NLP techniques in order to identify patients.

(4) Accommodate (and potentially combine) all definitions types ii



1	case assignment	rx_t2dm_med-abnormal_lab	A case is identified in the presence of an abnormal lab value (defined as one of three different exacerbations in blood sugar level) AND if medication for this type of diabetes has been prescribed.	boolean

Input		Output		
dm_cohort-abnormal_lab	Potential t2dm cases, with abnormal lab results identified	dm_cases-case_assignment_1	t2dm cases, as identified by first case assignment rule	csv

rx_t2dm_med-abnormal_lab.knwf	knime	-File=
rx_t2dm_med-abnormal_lab.py	python	-

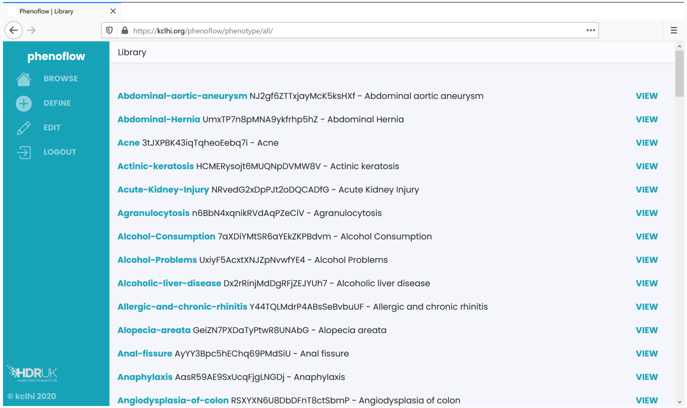
```
...
 if row["t1dm_dx_cnt"] == "0"
 and row["t2dm_dx_cnt"] == "0"
 and "t2dm_rx_dts" in csv_reader.fieldnames
 and "abnormal_lab" in csv_reader.fieldnames
 and row["abnormal_lab"] == "1":
   row["t2dm"] = "CASE";
...
```

Figure 3: Individual step of T2DM **logic-based** Phenoflow definition and implementation units.

Phenotype tooling

Phenoflow web architecture

The Phenoflow model is complemented by a **web architecture** that accentuates its benefits.

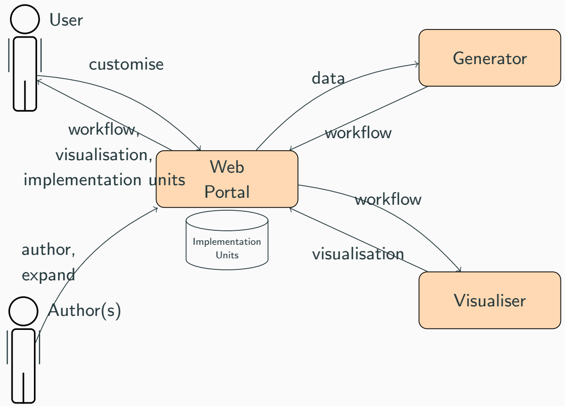


Abdominal-aortic-aneurysm NJ2gf6ZTTxjayMcK5ksHXf - Abdominal aortic aneurysm	VIEW
Abdominal-Hernia UmxTP7n8pMNA9ykfrhp5hZ - Abdominal Hernia	VIEW
Acne 3tJXPBK43iqTqheoEebq7i - Acne	VIEW
Actinic-keratosis HCMERysojt6MUQNpDVMW8V - Actinic keratosis	VIEW
Acute-Kidney-Injury NRvedG2xDpPJt2oDQCADfG - Acute Kidney Injury	VIEW
Agranulocytosis n6BbN4xqnikRVdAqPZeCiV - Agranulocytosis	VIEW
Alcohol-Consumption 7aXDiYMtSR6aYEkZKPBdvm - Alcohol Consumption	VIEW
Alcohol-Problems UxiyF5AcxtXNJZpNvwfYE4 - Alcohol Problems	VIEW
Alcoholic-liver-disease Dx2RinjMdDgRFjZEJYUh7 - Alcoholic liver disease	VIEW
Allergic-and-chronic-rhinitis Y44TQLMdrP4ABsSeBvbuUF - Allergic and chronic rhinitis	VIEW
Alopecia-areata GeiZN7PXDaTyPtwR8UNAbG - Alopecia areata	VIEW
Anal-fissure AyYY3Bpc5hEChq69PMdSiU - Anal fissure	VIEW
Anaphylaxis AasR59AE9SxUcqFjgLNGDj - Anaphylaxis	VIEW
Angiodysplasia-of-colon RSXYXN6U8DbDFnT8ctSbmP - Angiodysplasia of colon	VIEW

Chapman, Martin, et al. "Phenoflow: A microservice architecture for portable workflow-based phenotype definitions." AMIA, 2021.

Phenoflow web architecture

The Phenoflow model is complemented by a **web architecture** that accentuates its benefits.



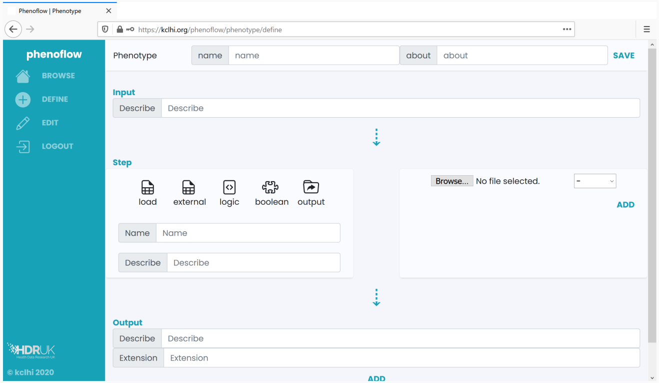
Chapman, Martin, et al. "Phenoflow: A microservice architecture for portable workflow-based phenotype definitions." AMIA, 2021.

Aside: Microservice architectures

We separate any system into individual services to ensure **scalability** (service replication), **resilience** (service indepedence), **technology heterogeneity** (allowing different people to use their favourite languages), **composability** (enabling reuse) and **ease of deployment**.

Authoring i

1. Author a **new** definition under the model.
   1.1 Represent an **existing** definition in a standard way.



Phenotype	
name	name
about	about

Input

Describe	Describe

Step

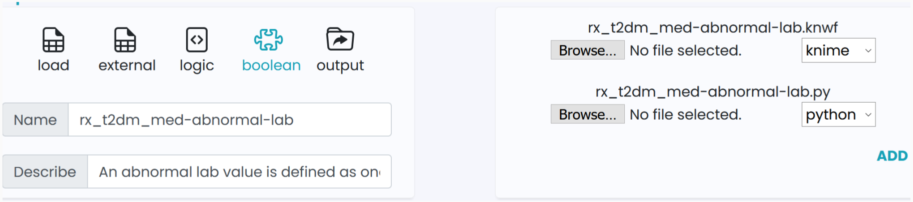
Name	Name
Describe	Describe

Output

Describe	Describe
Extension	Extension

Authoring ii

2. **Upload** implementation units for each step in the model.
   2.1 The now modular nature of the definition provides a **template** for development.
   2.2 Alternatively, allows existing implementations developed by users to be **reused** in a standard context.



Name	rx_t2dm_med-abnormal-lab
Describe	An abnormal lab value is defined as on

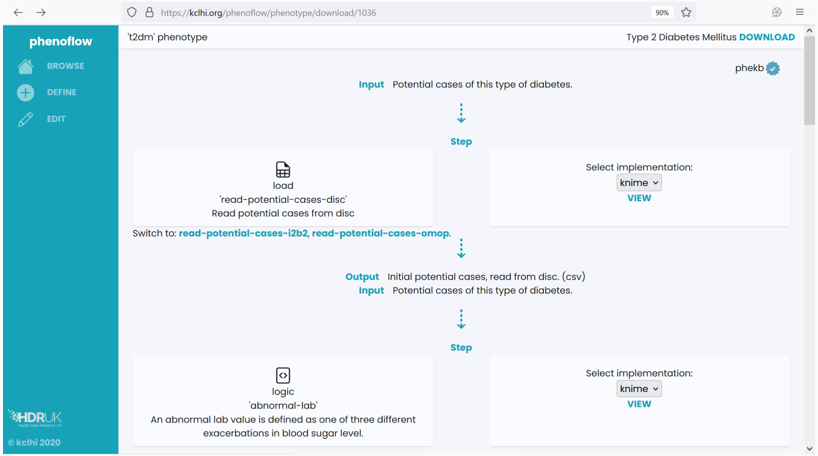
rx_t2dm_med-abnormal-lab.knwf
Browse... No file selected. knime

rx_t2dm_med-abnormal-lab.py
Browse... No file selected. python

3. Users can upload one or more implementations for each step in their **own** definitions, or the definitions **created by others**.

Execution i

1. Export as **CWL workflows**.



https://kclhi.org/phenoflow/phenotype/download/1036

't2dm' phenotype

Type 2 Diabetes Mellitus **DOWNLOAD**

phekb

Input Potential cases of this type of diabetes.

Step

load
'read-potential-cases-disc'
Read potential cases from disc

Select implementation: knime
VIEW

Switch to: **read-potential-cases-i2b2**, **read-potential-cases-omop**.

Output Initial potential cases, read from disc. (csv)
Input Potential cases of this type of diabetes.

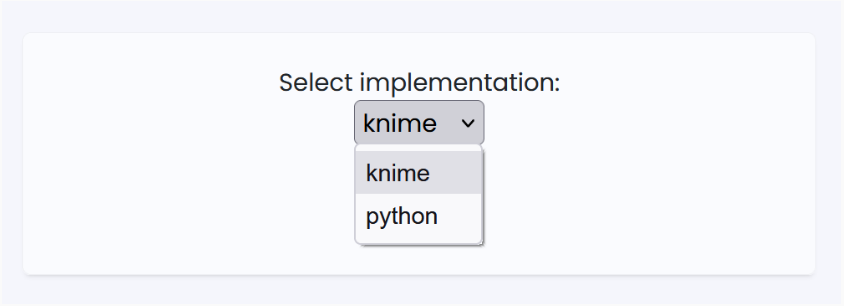
Step

logic
'abnormal-lab'
An abnormal lab value is defined as one of three different exacerbations in blood sugar level.

Select implementation: knime
VIEW

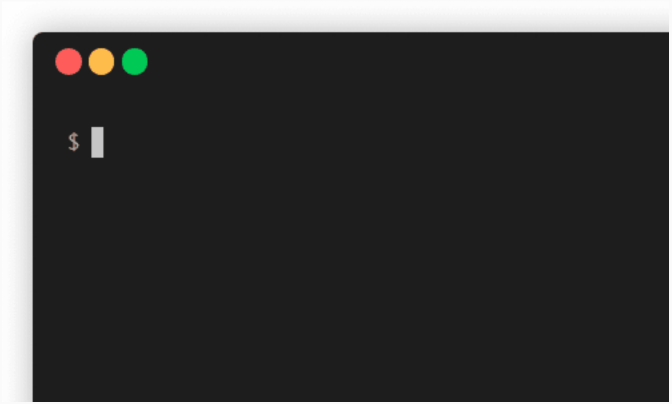
Execution ii

1.1 Can be edited by technical users who, if there are multiple uploads for each step, can **configure** how each step is implemented prior to download, in order to provide them with **familiar languages** with which to work.



Execution iii

1.2 Can simply be executed **out of the box** by non-technical users[2]

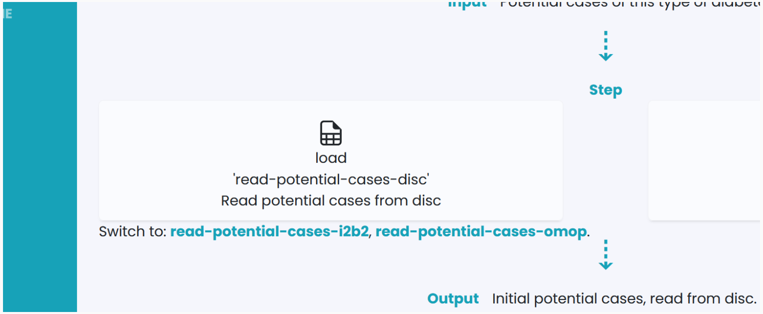


```
$ 
```

[2]Currently requires the command line, a GUI executor is forthcoming!

Execution iv

2. Pick a **connector** to be the first step in the workflow, depending on the format of the dataset you are targeting.
   2.1 **Credentials** for data stores are entered locally.



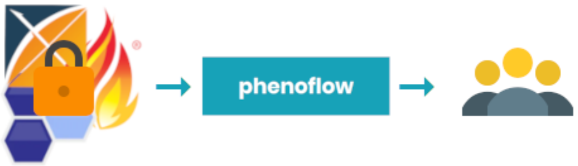
Step

load
'read-potential-cases-disc'
Read potential cases from disc

Switch to: read-potential-cases-i2b2, read-potential-cases-omop.

Output Initial potential cases, read from disc.

Execution v



Applications

Initial evaluation

First showed **portability improvements** in terms of clinical knowledge requirements and programming expertise using the *Knowledge conversion, clause Interpretation, and Programming* (KIP) phenotype portability scoring system (Shang et al., JBI, 2019.)

	Knowledge	Clause	Programming	Total
Traditional Code	0	2	2	4
Phenoflow Code	0	0	0	0
Traditional Logic	1	1	2	4
Phenoflow Logic	0	1	0	1

Table 2: KIP scores indicating the portability of traditional code-based (COVID-19) and logic-based (Type 2 Diabetes) phenotype definitions and their Phenoflow counterparts.

Clinical trials i

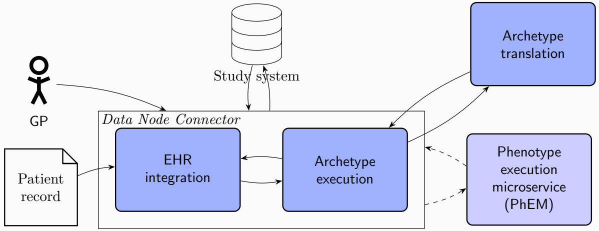
REST
the runny ear study

Recruitment in the REST clinical trial (AOMd) was handled using the **TRANSFoRm** e-source trial platform.

In the original trial, archetype-based criteria were translated to **concrete implementations** (e.g. XPath queries) by TRANSFoRm in order to determine a patient's eligibility from their EHR.

Clinical trials ii

We developed a new service (**PhEM**) that instead enables the execution of a computable phenotype against an EHR in order to identify eligible patients at the point-of-care.



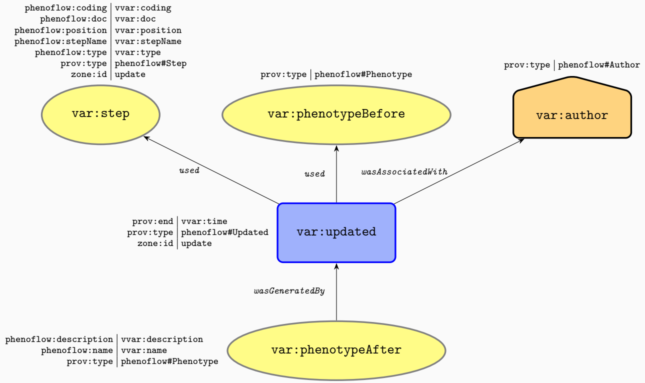
The use of PhEM was shown to increase recruitment accuracy.

Chapman, Martin, et al. "Using Computable Phenotypes in Point-of-Care Clinical Trial Recruitment". MIE, 2021.

Provenance

A **reverse application**: connected Phenoflow with the **Data Provenance Template** server, a piece of software that holds structured fragments of **provenance**.

These fragments record the evolution of the data (definitions) within Phenoflow, as they are edited by users, improving **validity**, **intelligibility** and **reproducibility**:



phenoflow:coding | vvar:coding
phenoflow:doc | vvar:doc
phenoflow:position | vvar:position
phenoflow:stepName | vvar:stepName
phenoflow:type | vvar:type
prov:type | phenoflow#Step
zone:id | update
var:step
prov:type | phenoflow#Phenotype
var:phenotypeBefore
prov:type | phenoflow#Author
var:author
used
used
wasAssociatedWith
prov:end | vvar:time
prov:type | phenoflow#Updated
zone:id | update
var:updated
wasGeneratedBy
phenoflow:description | vvar:description
phenoflow:name | vvar:name
prov:type | phenoflow#Phenotype
var:phenotypeAfter

Fairweather, Elliot, et al. "A delayed instantiation approach to template-driven provenance for electronic health record phenotyping". IPAW, 2020.

The future

Authoring vs. parsing

“A model is only useful if it’s **used**”.

A challenging task to expect the adoption of a **single model**.

Alternative approach: **parse** the definitions developed by others (e.g. those represented in other libraries), represent them within Phenflow, and then provide them for use.

Parsing

Grab **data** relating to a definition (codelist, drug list, keywords, more complex logic).

↓

Determine where **key information** is within this data (e.g. a ‘conceptid’ column in a codelist CSV).

↓

Populate the **information required** for the Phenoflow model automatically (e.g. step structure – easier for something like a codelist; may require some **intervention** with more complex logic).

↓

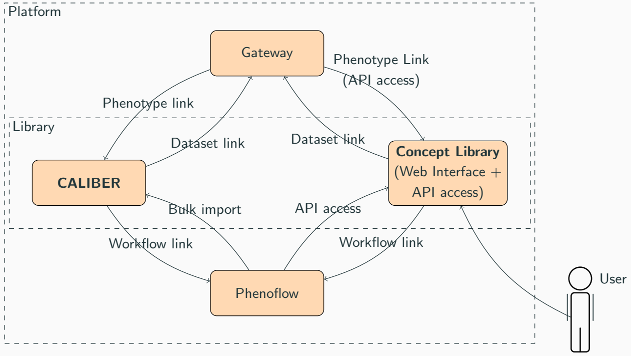
Automatically **generate** implementation units for each step.

↓

Add these to the Phenoflow library ready for **download**.

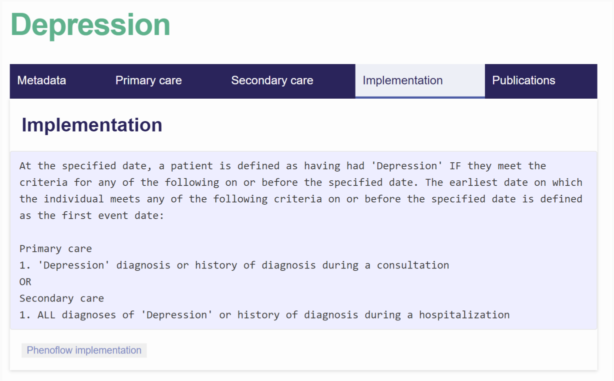
Parsing example – HDR UK national phenomics resource i

Have **imported**, **standardised** and provided **implementations** for ∼300 existing definitions as a part of the HDR phenomics resource:



Parsing example – HDR UK national phenomics resource ii

These are now **directly linked to** from CALIBER, and from the soon to be release **HDR UK Phenotype Portal**:



Implementation

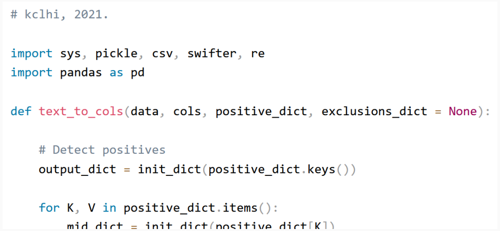
```
At the specified date, a patient is defined as having had 'Depression' IF they meet the
criteria for any of the following on or before the specified date. The earliest date on which
the individual meets any of the following criteria on or before the specified date is defined
as the first event date:

Primary care
1. 'Depression' diagnosis or history of diagnosis during a consultation
OR
Secondary care
1. ALL diagnoses of 'Depression' or history of diagnosis during a hospitalization
```

Parsing example - KCLHI NLP Phenotypes

The Health Informatics group at King's have derived a set of **inclusion** (and exclusion – yet to be modelled) **keywords** for a range of conditions.

Steps of model, and other required information, **generated** based on this data.

Example implementation provided and used as part of parsing process to generate implementation units.



```python
# kclhi, 2021.

import sys, pickle, csv, swifter, re
import pandas as pd

def text_to_cols(data, cols, positive_dict, exclusions_dict = None):

    # Detect positives
    output_dict = init_dict(positive_dict.keys())

    for K, V in positive_dict.items():
        mid_dict = init_dict(positive_dict[K])
```

Other future work i

- Always interested in **parsing definitions from new sources**.
- Publish more implementations for complex disease-specific phenotypes, e.g. long covid (LOCOMOTION; phenotypes from NW London GP records) and stroke (KCL NIHR; phenotypes from SLSR).
- Increase the library of **workflow modules** (e.g. types of dataset connectors) ready for download and use.
- Automatic **data conversion** to enable use of different implementation techniques on same dataset, e.g. conversion from CSV to DB to allow use of SQL scripts.

Thank you!

Welcome to visit Phenoflow itself: **http://kclhi.org/phenoflow**.

View the architecture on Github: **https://github.com/kclhi/phenoflow**.

Publications mentioned: **https://martinchapman.co.uk/publications/pheno**.

Contact: **@martin_chap_man**.