Type 2 Diabetes Mellitus Electronic Medical Record Case and Control Selection Algorithms

Jennifer Allen Pacheco
NUgene Informatics Lead
japacheco@northwestern.edu

Will Thompson
Senior Research Associate
wkt@northwestern.edu

August 19, 2011

1 Overview

This document describes the Northwestern University Type 2 diabetes mellitus (T2DM) algorithms for extracting both T2DM cases and T2DM controls from the electronic medical record (EMR). There are two main parts to this document. The first part (Section 2) provides descriptions of the input data elements to be extracted from the EMR, flowcharts, and pseudo-code descriptions of the algorithms. The second part (Section 3) is an installation guide for executable workflows that implement the T2D case and control selection algorithms. These worfklows are based on the Konstanz Information Miner (KNIME) data analysis platform.[1]

2 Algorithm Descriptions

The case and control selection algorithms require certain patient-level data elements to be extracted from the EMR. This information includes diagnoses, lab results, medication orders, and physician encounter dates. Lists of codes that satisfy various algorithm requirements (including ICD-9 codes, LOINC codes, and RxNorm codes) are provided in tabular form in Appendix A. Additionally, Section 3.1 contains a translation of these data elements into data dictionaries for input into the KNIME workflow implementations.[2]

[1] Questions about the core algorithms should be sent to japacheco@northwestern.edu, while questions about the executable KNIME workflows should be sent to wkt@northwestern.edu.

[2] See also the T2D study in the eleMAP online tool. This study contains data elements that were used in a T2D genome-wide association study (GWAS), using a patient cohort derived from the EMR-based algorithm described in this document.

2.1 T2DM Case Selection Algorithm Logic

For the T2D case selection algorithm, the following data elements are required:

1. Counts of T1DM ICD-9 code assignment dates by diagnostic source (Table 3)
2. Counts of T2DM ICD-9 code assignment dates by diagnostic source (Table 4)
3. T1DM medications (i.e., Insulin & Symlin) order or prescription dates – at least the earliest date of Rx (Table 5)
4. T2DM medications order or prescription dates – at least the earliest date of Rx (Table 6)
5. Fasting blood glucose lab values – at least the maximum value (Table 7)
6. Random blood glucose lab values – at least the maximum value (Table 7)
7. HBA1c lab values – at least the maximum value (Table 7)

For the T2D case selection algorithm, the following definitions apply:

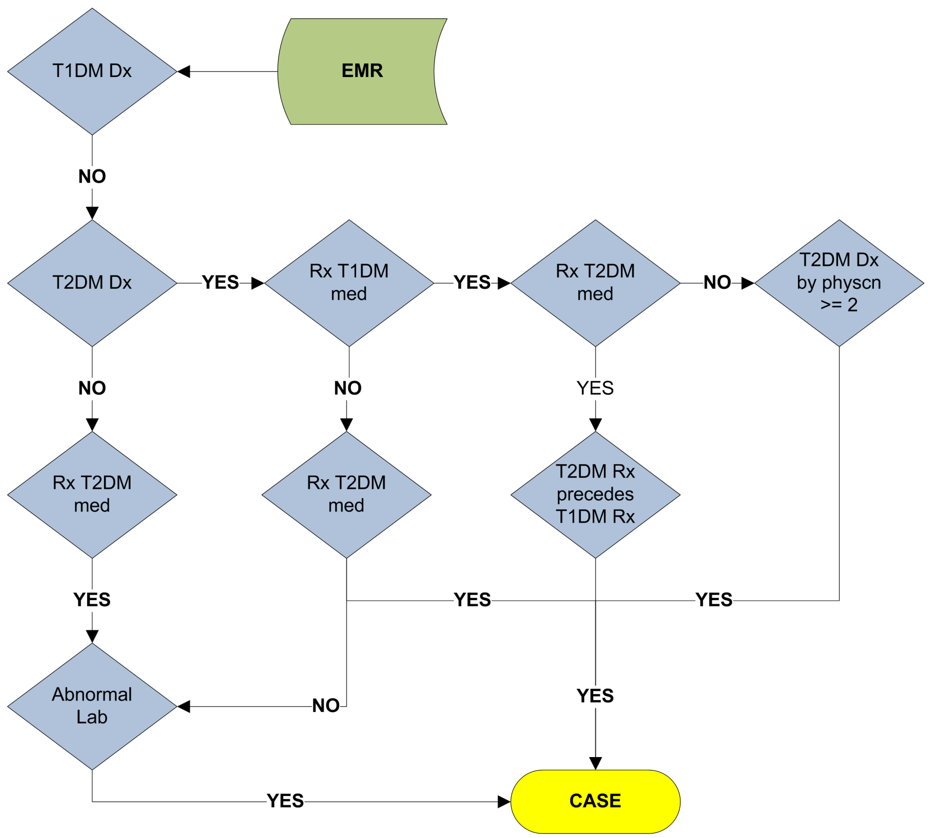
- *Abnormal lab* – An abnormal lab value is defined as one of the following:
  - Random glucose $>$ 200 mg/dl
  - Fasting glucose $\geq$ 125 mg/dl
  - Hemoglobin A1c $\geq$ 6.5%
- *Physician entered diagnosis* – A physician entered diagnosis code is one that is derived from encounter or problem list sources only (excludes billing codes).

A flowchart expressing the logic of the T2DM case selection algorithm is shown in Figure 1. There are five paths through this flowchart, and these five paths are translated into equivalent pseudo-code in Algorithm 1. This algorithm shows the top-level logic, with additional sub-procedures implementing the lower-level details. These sub-procedures are also expressed below, in terms of an SQL-like syntax that is linked to the ICD-9, LOINC, and RxNorm codes in Appendix A.[3]

[3] These are the abbreviations used in the following flowcharts and algorithms:

- DX → DIAGNOSIS
- RX → PRESCRIPTION
- PT → PATIENT
- DT → DATE
- CNT → COUNT

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



Algorithm 1 T2DM case selection algorithm. This algorithm takes a patient-level record (pt) as an argument, and returns the patient's case status ({CASE, UNKNOWN}) as result.

```
T2DM-CASE-SELECTION(pt)
    status = UNKNOWN

1   if T1DM-DX-DT-CNT(pt) == 0                          ⇐ Algorithm 2
    AND T2DM-DX-DT-CNT(pt) > 0                          ⇐ Algorithm 3
    AND T2DM-RX-DT(pt) ≠ NULL                           ⇐ Algorithm 4
    AND T1DM-RX-DT(pt) ≠ NULL                           ⇐ Algorithm 5
    AND T2DM-RX-DT(pt) < T1DM-RX-DT(pt)
        status = CASE
2   elseif T1DM-DX-DT-CNT(pt) == 0
    AND T2DM-DX-DT-CNT(pt) > 0
    AND T1DM-RX-DT(pt) == NULL
    AND T2DM-RX-DT(pt) ≠ NULL
        status = CASE
3   elseif T1DM-DX-DT-CNT(pt) == 0
    AND T2DM-DX-DT-CNT(pt) > 0
    AND T1DM-RX-DT(pt) == NULL
    AND T2DM-RX-DT(pt) == NULL
    AND ABNORMAL-LAB(pt) == TRUE                        ⇐ Algorithm 6
        status = CASE
4   elseif T1DM-DX-DT-CNT(pt) == 0
    AND T2DM-DX-DT-CNT(pt) == 0
    AND T2DM-RX-DT(pt) ≠ NULL
    AND ABNORMAL-LAB(pt) == TRUE
        status = CASE
5   elseif T1DM-DX-DT-CNT(pt) == 0
    AND T2DM-DX-DT-CNT(pt) > 0
    AND T1DM-RX-DT(pt) ≠ NULL
    AND T2DM-RX-DT(pt) == NULL
    AND T2DM-PHYSCN-DX-DT-CNT(pt) ≥ 2                   ⇐ Algorithm 7
        status = CASE

    return status
```

Algorithm 2 Count of distinct dates of T1DM DX (called by Algorithm 1)

```
T1DM-DX-DT-CNT(pt)
    count =
        select COUNT-DISTINCT-DT(records)
        from dx-table
        where
            dx-table.pt == pt
            AND dx-table.icd-9-code ∈ {...}    ⇐ Table 3
    return count
```

Algorithm 3 Count of distinct dates of T2DM DX (called by Algorithm 1)

```
T2DM-DX-DT-CNT(pt)
    count =
        select COUNT-DISTINCT-DT(records)
        from dx-table
        where
            dx-table.pt == pt
            AND dx-table.icd-9-code ∈ {...}    ⇐ Table 4
    return count
```

Algorithm 4 First date of Rx for T2DM medication (called by Algorithm 1)

```
T2DM-RX-DT(pt)
    dt =
        select FIRST-DT(records)
        from rx-table
        where
            rx-table.pt == pt
            AND rx-table.rxnorm-code ∈ {...} ⇐ Table 6
    return dt
```

Algorithm 5 First date of Rx for T1DM medication (called by Algorithm 1)

T1DM-RX-DT(*pt*)
    dt =
        select FIRST-DT(*records*)
        from *rx-table*
        where
            rx-table . pt == *pt*
            AND *rx-table . rxnorm-code* ∈ {. . .} ⇐ Table 5
    return *dt*

Algorithm 6 Check for abnormal lab (called by Algorithm 1)

ABNORMAL-LAB(*pt*)
    abnormal-lab = FALSE
    lab-results =
        select *records*
        from *labs-table*
        where
            labs-table . pt == *pt*
            AND *labs-table . loinc-code* ∈ {. . .} ⇐ Table 7
    for each *lab* ∈ *lab-results*
        if *lab . type* == RANDOM-GLUCOSE
            AND *lab . value* ≥ 200 // (mg/dl)
        OR *lab . type* == FASTING-GLUCOSE
            AND *lab . value* ≥ 125 // (mg/dl)
        OR *lab . type* == HBA1C
            AND *lab . value* ≥ 6.5 // (percent)
            abnormal-lab = TRUE
    return *abnormal-lab*

Algorithm 7 Count of distinct dates of T2DM DX made by a physician (called by Algorithm 1)

```
T2DM-PHYSCN-DX-DT-CNT(pt)
    count =
        select COUNT-DISTINCT-DT(records)
        from dx-table
        where
            dx-table.pt == pt
            AND dx-table.source ∈ {ENCOUNTER, PROBLEM-LIST}
            AND dx-table.icd-9-code ∈ {...}    ⇐ Table 4
    return count
```

2.2 T2DM Control Selection Algorithm Logic

For the T2D control selection algorithm, the following data elements are required:

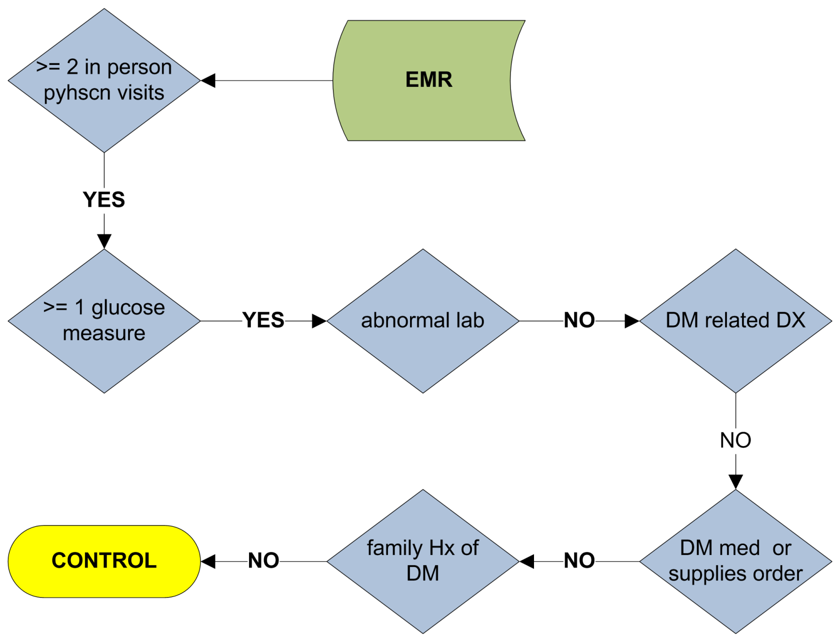
1. Counts of ICD-9 codes related to diabetes (Table 9)
2. Fasting blood glucose lab values – at least the maximum value (Table 7)
3. Random blood glucose lab values – at least the maximum value (Table 7)
4. HBA1c lab values – at least the maximum value (Table 7)
5. Diabetes family history – could be supplemented with self-reported data from a questionnaire
6. T1DM medications (i.e., Insulin & Symlin) order or prescription dates – at least the earliest date of Rx (Table 5)
7. T2DM medications order or prescription dates – at least the earliest date of Rx (Table 6)
8. Diabetes medical supply orders (Table 8)
9. Count of dates the patient had face-to-face outpatient clinic encounters (in-person visits with a clinician)

For the control selection algorithm, the following definition applies:

- *Abnormal lab*: abnormal lab values include:
  - Random glucose $>$ 110 mg/dl
  - Fasting glucose $\geq$ 110 mg/dl
  - Hemoglobin A1c $\geq$ 6.0%

A flowchart expressing the logic of the T2DM control selection algorithm is shown in Figure 2. There is only one path through this flowchart, and this path is translated into equivalent pseudo-code in Algorithm 8. This algorithm shows the top-level logic, with additional sub-procedures implementing the lower-level details. These sub-procedures are also expressed below, in terms of an SQL-like syntax that is linked to the ICD-9, LOINC, and RxNorm codes in Appendix A.

Figure 2: Algorithm for identifying T2DM controls in the EMR.



Algorithm 8 TT2DM control selection algorithm. This algorithm takes a patient-level record (pt) as an argument, and returns the patient's control status ({CONTROL, UNKNOWN}) as result.

T2DM-CONTROL-SELECTION(pt)

status = UNKNOWN

if DM-DX-DT-CNT(pt) == 0 ⇐ Algorithm 9
AND GLUCOSE-LAB-EXISTS(pt) == TRUE ⇐ Algorithm 10
AND ABNORMAL-LAB(pt) == FALSE ⇐ Algorithm 11
AND ENCTRS-DT-CNT(pt) ≥ 2 ⇐ Algorithm 12
AND DM-MEDS-SUPPLIES-RX-DT-CNT(pt) == 0 ⇐ Algorithm 13
AND FAM-HIST-OF-DM(pt) == FALSE ⇐ Algorithm 14
status = CONTROL

return *status*

Algorithm 9 Count of distinct dates of DM-related DX (called by Algorithm 8)

```
DM-DX-DT-CNT(pt)
    count =
        select COUNT-DISTINCT-DT(records)
        from dx-table
        where
            dx-table.pt == pt
            AND dx-table.icd-9-code ∈ {...}     ⇐ Table 9
    return count
```

Algorithm 10 Check for glucose lab performed (called by Algorithm 8)

```
GLUCOSE-LAB-EXISTS(pt)
    glucose-lab-exists = FALSE
    lab-results =
        select records
        from labs-table
        where
            labs-table.pt == pt
            AND labs-table.loinc-code ∈ {...}   ⇐ Table 7 (glucose only)
    if lab-results.count > 0
        glucose-lab-exists = TRUE
    return glucose-lab-exists
```

Algorithm 11 Check for abnormal lab (called by Algorithm 8)

```
ABNORMAL-LAB(pt)
    abnormal-lab = FALSE
    lab-results =
        select records
        from labs-table
        where
            labs-table.pt == pt
            AND labs-table.loinc-code ∈ {...}   ⇐ Table 7

    for each lab ∈ lab-results
        if lab.type == RANDOM-GLUCOSE
            AND lab.value ≥ 110 // (mg/dl)
        OR lab.type == FASTING-GLUCOSE
            AND lab.value ≥ 110 // (mg/dl)
        OR lab.type == HBA1C
            AND lab.value ≥ 6.0 // (percent)
            abnormal-lab = TRUE
    return abnormal-lab
```

Algorithm 12 Count of distinct dates for in-person office encounters with a physician (called by Algorithm 8)

```
ENCTRS-DT-CNT(pt)
    count =
        select COUNT-DISTINCT-DT(records)
        from enctrs-table
        where
            enctrs-table.pt == pt
            AND enctrs-table.type == OFFICE
    return count
```

Algorithm 13 Count DM-related medications and supplies by distinct Rx date (called by Algorithm 8)

```
DM-MEDS-SUPPLIES-RX-DT-CNT(pt)
    count =
        select COUNT-DISTINCT-DT(records)
        from rx-table
        where
            rx-table.pt == pt
            AND
                rx-table.rxnorm-code ∈ {...}          ⇐ Table 5
                OR rx-table.rxnorm-code ∈ {...}       ⇐ Table 6
                OR rx-table.rxnorm-code ∈ {...}       ⇐ Table 8
    return count
```

Algorithm 14 Check for family history of DM (called by Algorithm 8)

```
FAM-HIST-OF-DM(pt)
    fam-hist-results =
        select records
        from fam-hist-table
        where
            fam-hist-table.pt == pt
            AND
                fam-hist-table.t1dm == TRUE
                OR fam-hist-table.t2dm == TRUE
    if fam-hist-results.count > 0
        return TRUE
    else return FALSE
```

3 KNIME workflow

This section describes installation of executable workflows that implement the case and control algorithms described in Section 2. These workflows are executed inside of the Konstanz Information Miner (KNIME) data analysis platform. The workflows take as input comma-separated value (csv) files, with each row corresponding to a patient (for examples, see the sample input files `dm_potential_cases.csv` and `dm_potential_controls.csv`).

3.1 Data Dictionaries

Each row of input data consists of a set of patient-level variables. We present here the data dictionaries that describe these patient-level input variables. The columns of the dictionaries specify each variable's name, type, and range of possible values. The next column specifies whether or not missing values are permitted, and if so, what the default value of the variable is.[4] The final column refers (where appropriate) to the table in Appendix A where corresponding code values for the variable can be found.

Table 1: Input variables to the T2DM case selection KNIME workflow

Name	Type	Range	Missing (def.)	Ref.
pat_id (unique)	INTEGER	$n \geq 1$	FALSE	NA
t1dm_dx_cnt	INTEGER	$n \geq 0$	TRUE (0)	Table 3
t2dm_dx_cnt	INTEGER	$n \geq 0$	TRUE (0)	Table 4
t2dm_physcn_dx_cnt	INTEGER	$n \geq 0$	TRUE (0)	Table 4
t1dm_rx_dt	STRING	`yyyy-mm-dd`	TRUE (NULL)	Table 5
t2dm_rx_dt	STRING	`yyyy-mm-dd`	TRUE (NULL)	Table 6
max_fast_gluc_lab_val	FLOAT	$n \geq 0.0$	TRUE (NULL)	Table 7
max_rndm_gluc_lab_val	FLOAT	$n \geq 0.0$	TRUE (NULL)	Table 7
max_hba1c_lab_val	FLOAT (%)	$0.0 \leq n \leq 100.0$	TRUE (NULL)	Table 7

[4] The default value is automatically inserted for a variable when it is missing a specified value.

Table 2: Input variables to the T2DM control selection KNIME workflow

Name	Type	Range	Missing (def.)	Ref.
pat_id (unique)	INTEGER	$n \geq 1$	FALSE	NA
fam_hist_of_dm	INTEGER	$n \in \{0, 1\}$	TRUE (0)	NA
enctrs_cnt	INTEGER	$n \geq 0$	TRUE (0)	Table 3
max_fast_gluc_lab_val	FLOAT	$n \geq 0.0$	TRUE (NULL)	Table 7
max_rndm_gluc_lab_val	FLOAT	$n \geq 0.0$	TRUE (NULL)	Table 7
max_hba1c_lab_val	FLOAT (%)	$0.0 \leq n \leq 100.0$	TRUE (NULL)	Table 7
dm_dx_cnt	INTEGER	$n \geq 0$	TRUE (0)	Table 9
dm_med_supplies_cnt	INTEGER	$n \geq 0$	TRUE (0)	Table 5, Table 6, Table 8

3.2 Installation and Execution

1. Download and install KNIME (version 2.4 or later). The KNIME website contains installation instructions, as well as tutorials.

2. Download the T2D case and control workflows, which are contained in a single zip file: `T2D-workflows.zip`. Don't unzip the file.

3. Download the two sample input files for the workflows: `dm_potential_cases.csv` and `dm_potential_controls.csv`.

4. Start KNIME. On start-up, you will see an empty workspace similar to the screenshot in Figure 3.

5. Select File ⇒ Import KNIME workflow... The resulting pop-up window is shown in Figure 4. Click on the Select archive file: radio button, and navigate to your local copy of the `T2D-workflows.zip` file. Click on the Finish button.

6. Double-click on the Diabetes-Case-Assignment workflow to open it. Your workspace will now look similar to the screenshot in Figure 5.[5]

7. Double-click on the File Reader node in the workflow graph. You will see the pop-up window shown in Figure 6. Click on the Browse... button and navigate to your local copy of the `dm_potential_cases.csv` file. Your pop-up window should look like the one in Figure 7. Make sure that the read row IDs box is unchecked, while the read column headers box is checked. Click on the OK button to close the window.

8. Double-click on the CSV Writer node in the workflow graph. Click on the Browse... button and navigate to a directory of your choosing where the output file `dm_cases.csv` will be generated. Click on the OK button to close the window.

9. The workflow is now ready to execute. Click on the green button with the double arrow at the toolbar at the top, or enter Shift+F7 on the keyboard. If the nodes of the workflow have already been executed[6], then first select all nodes (Control+A), right click, and select Reset.

   (a) The output file will be located in the directory that you chose in Step 8.

   (b) Right click on the Rule Engine node in the graphical view and select Classified Data in order to view the output of the algorithm (Figure 8).

   (c) Right click on the Histogram node in the graphical view and select View:Histogram View to get counts of the assignments that were made (Figure 9).

[5] All following steps apply also to the Diabetes-Control-Assignment workflow.

[6] An executed node will have a green indicator underneath it.

Figure 3: Step 4

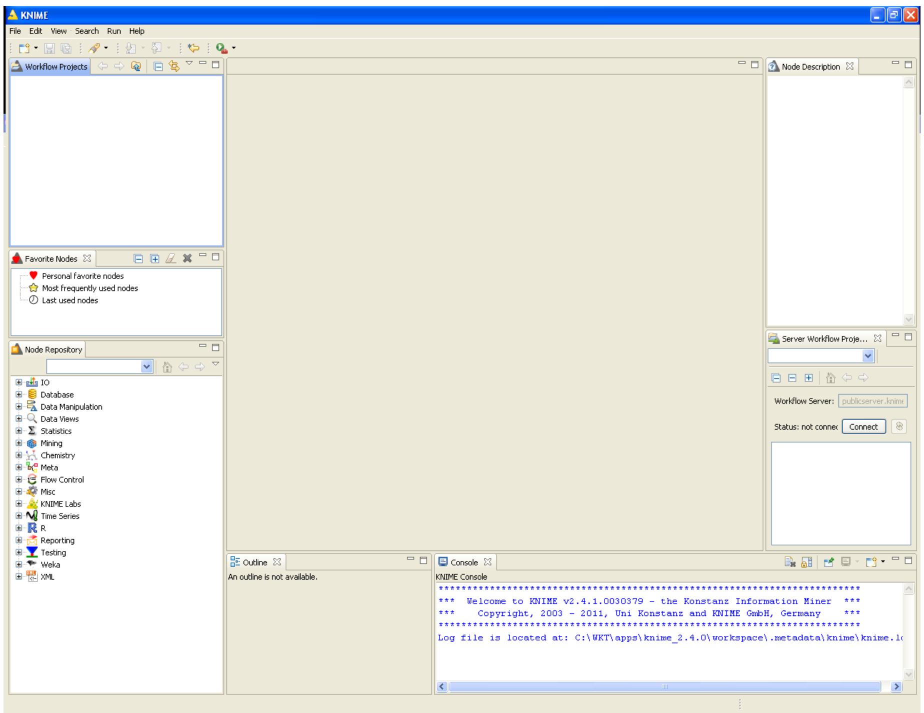


Figure 4: Step 5

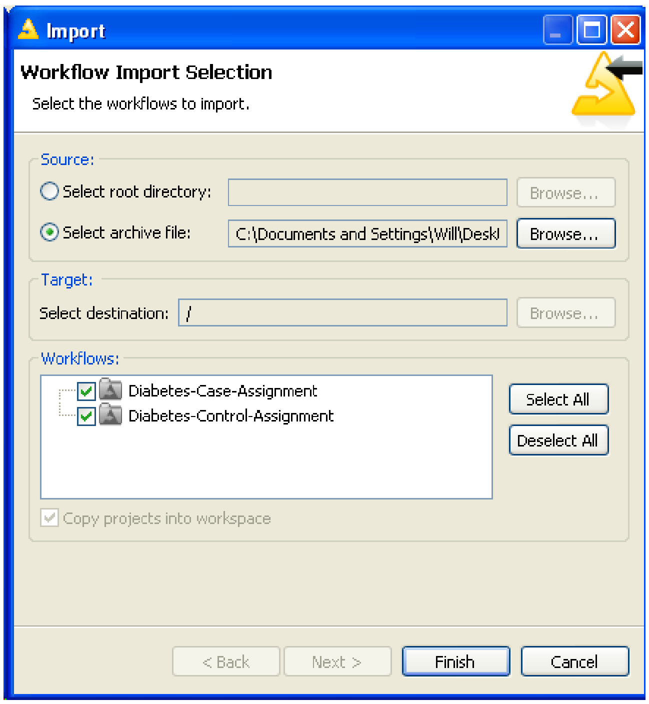


Figure 5: Step 6

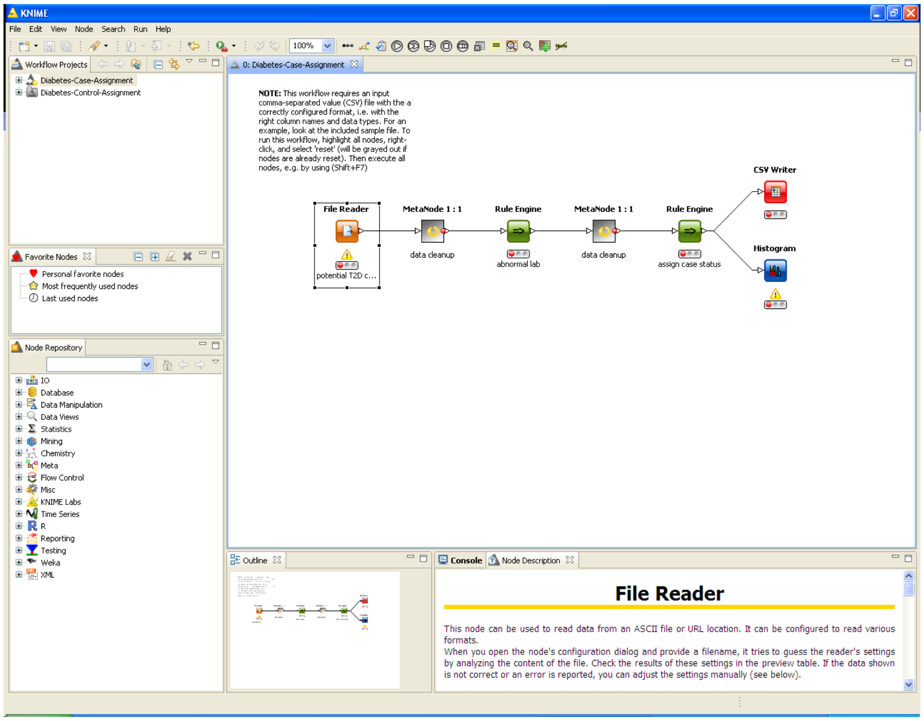


Figure 6: Step 7

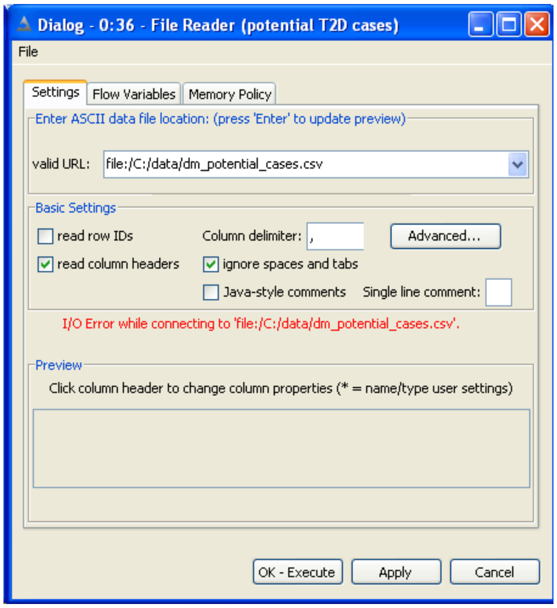


Figure 7: Step 7

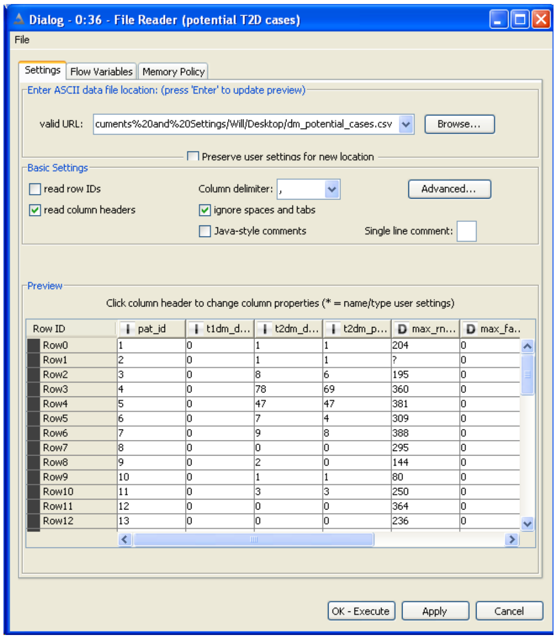


Figure 8: Step 9b

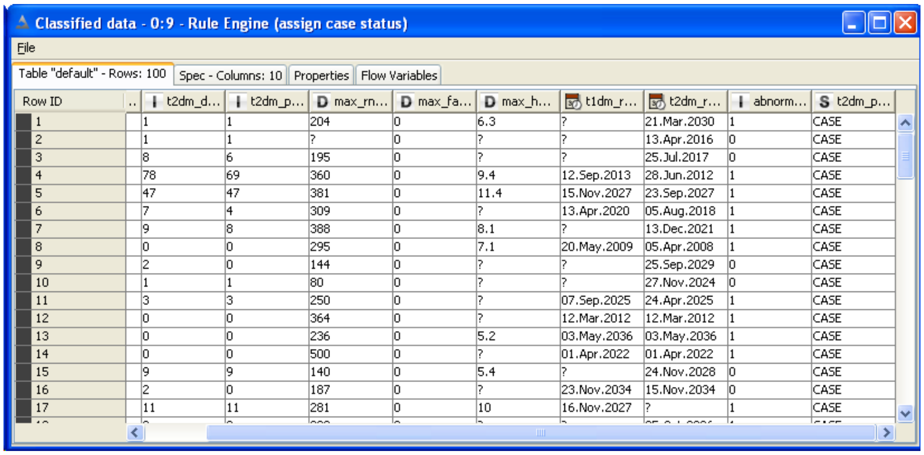
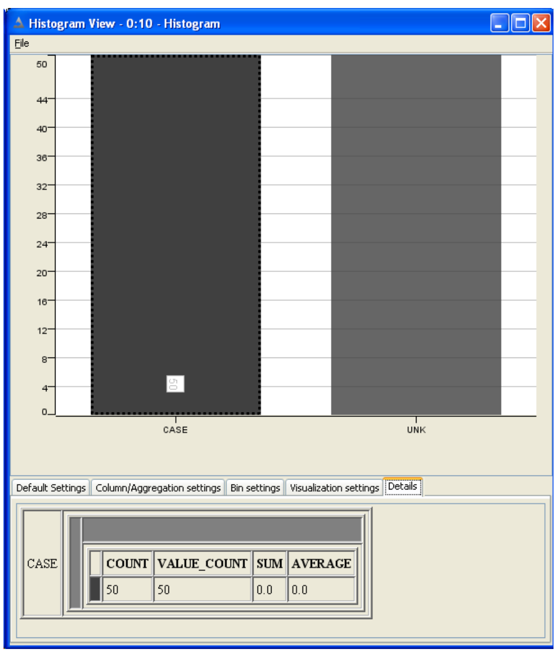


Figure 9: Step 9c



A Data Elements

Table 3: T1DM diagnosis codes. Used in Algorithm 1.

Description	ICD-9 code
Type 1 Diabetes	250.x1, 250.x3

Table 4: T2DM diagnosis codes. Used in Algorithm 1.

Description	ICD-9 code
Type 2 Diabetes	250.x0, 250.x2 (excl. 250.10, 250.12)

Table 5: T1DM medications. Used in Algorithm 1 and Algorithm 8.

Generic Name	Example Brand	RxNorm CUI (ingredient-level)
insulin		139825, 274783, 314684, 352385, 400008, 51428, 5856, 86009
pramlintide	Symlin	139953

Table 6: T2DM medications. Used in Algorithm 1 and Algorithm 8.

Generic Name	Example Brand	RxNorm CUI (ingredient-level)
acetohexamide	Dymelor	173
tolazamide	Tolinase	10633
chlorpropamide	Diabinese	2404
glipizide	Glucotrol	4821
glipizide	Glucotrol XL	217360
glyburide	Micronase, Glynase, Diabeta	4815
glimepiride	Amaryl	25789
repaglinide	Prandin	73044
nateglinide	Starlix	274332
metformin	Glucophage	6809
rosiglitazone	Avandia	84108
pioglitazone	ACTOS	33738
troglitazone	Rezulin	72610
acarbose	Precose	16681
miglitol	Glyset	30009
sitagliptin	Januvia	593411
exenatide	Byetta	60548

Table 7: Diabetes mellitus lab codes. Used in Algorithm 1 and Algorithm 8

Description	LOINC code
Fasting glucose	1558-6
Random glucose	2339-0, 2345-7
Hemoglobin A1C	4548-4, 17856-6, 4549-2, 17855-8

Table 8: Diabetes medical supplies. Used in Algorithm 8.

Description	Source Vocab.	RxNorm CUI (ingredient-level)
Blood-glucose meters & sensors	NDDF	126958, 412956, 412959, 637321, 668291, 668370, 686655, 692383, 748611, 880998, 881056
	VANDF	751128
Insulin syringes	RxNorm	847187, 847191, 847197, 847203, 847207, 847211, 847230, 847239, 847252, 847256, 847259, 847263, 847278, 847416, 847417
	NDDF	806905, 806903, 408119

Table 9: Diabetes mellitus diagnosis codes. Used in Algorithm 8.

Description	ICD-9 code
Diabetes mellitus (T1 & T2)	250.xx
Impaired fasting glucose	790.21
Impaired oral glucose tolerance test	790.22
Abnormal glucose not otherwise spec.	790.2, 790.29
Abnormal glucose during pregnancy	648.8x
Gestational diabetes	648.0x
Glycosuria	791.5
Dysmetabolic syndrome X	277.7
Family history of diabetes mellitus	V18.0
Screening for diabetes mellitus	V77.1