**Algorithm Specification Sheet - Alinity IA Pipettor FE Pressure**

**References**

APLM D000029930 Rev A

**Summary**

To implement a Predictive Health Monitoring (PHM) algorithm for the Alinity-i analyzers that will detect a degrading pipettor pressure transducer or loss-of-connection to the controller board and associated loss of pressure monitoring before the customer begins experiencing an increase in pipetting errors and/or erratic results due to inaccurate pipetting.

|  |  |
| --- | --- |
| PHN Descriptor | Alinity IA Pipettor FE Pressure |
| PHN Experience Code / Name | CC11 PHN\_Alinity\_IA; PHN\_R1 Pipettor; FE Pressure  CC21 PHN\_Alinity\_IA; PHN\_R2 Pipettor; FE Pressure  CCJ1 PHN\_Alinity\_IA; PHN\_Sample Pipettor; FE Pressure |
| PHN KM Article Number/ Name | K82525012 PHN\_IA: Front End (FE) Pressure |
| Skill Level | 2- Advanced Service |
| Always On Package | Always On 01DP5- 01, 02, 03, 09, 80, 83, 84, 89 |
| IDA Table | IDAQOWNER.ICQ\_PMEVENTS |
| IDA Table Fields | MODULESN, LOGDATE\_LOCAL, FRONTENDPRESSURE, PIPETTINGPROTOCOLNAME, PIPETTORMECHANISMNAME |
| Analysis Frequency | Daily |
| Data Required | Previous 1 day |
| Data Aggregation | None |
| Run Time Estimate | 6 seconds (1 day, 32 instruments, 8,465 rows, 1 algorithm flag) |
| Flag Criteria | FE Pressure greater than 27,000 or less than 21,000 |
| Probable Failure Modes | Pipettor pressure sensor.  Pipettor board and connections. |
| Suppression Experience Codes | None |
| Applicable Work Done Codes (WDC) | R1: BB1\* Fluid Aspiration, Dispense and Detection; R1 Probe; Multiple LLS / Pressure Monitoring Errors  R2: BC1\*: Fluid Aspiration, Dispense and Detection; R2 Probe; Multiple LLS / Pressure Monitoring Errors  Sample: B31\*: Fluid Aspiration, Dispense and Detection; Sample Probe; Multiple LLS / Pressure Monitoring Errors |

**Data Processing Steps**

|  |  |
| --- | --- |
| Data Processing Steps | |
| 1 | Query all data from the previous day for each pipettor/instrument combination (IDA Table: IDAQOWNER.ICQ\_PMEVENTS). |
| 2 | Unique pipettors will be identified by PIPETTORMECHANISMNAME and MODULESN. |
| 3 | Exclude aspirations with: PIPETTINGPROTOCOLNAME = "NonPipettingProtocol" |
| 4 | Exclude pipettors that have less than 10 aspirations. |
| 5 | Summarize the data by calculating the number of aspirations in which FE pressure (FRONTENDPRESSURE) is less than 21,000 or greater than 27,000. |
| 6 | Flag any pipettor/instrument combination with 2% or more of its aspirations having FE pressure less than 21,000 or greater than 27,000. Note that the 2% is necessary to exclude occluded probes where the problem is resolved through a system flush. |

**APPENDIX 1:** CC11 PHN\_Alinity\_IA; PHN\_R1 Pipettor; FE Pressure

**Algorithm Code**

SELECT

evals.MODULESN

FROM

(SELECT

PM.MODULESN,

COUNT(PM.PIPETTORMECHANISMNAME) AS ASPIRATIONS,

SUM(CASE WHEN PM.FRONTENDPRESSURE > 27000 OR PM.FRONTENDPRESSURE < 21000

THEN 1

ELSE 0

END) AS NUMFLAGS

FROM

IDAQOWNER.ICQ\_PMEVENTS PM

WHERE

PM.LOGDATE\_LOCAL >= TRUNC(SYSDATE) - 1

AND PM.LOGDATE\_LOCAL < TRUNC(SYSDATE)

AND PM.FRONTENDPRESSURE IS NOT NULL

AND PM.PIPETTINGPROTOCOLNAME != 'NonPipettingProtocol'

AND PM.PIPETTORMECHANISMNAME = 'Reagent1PipettorMechanism'

GROUP BY

PM.MODULESN

ORDER BY

PM.MODULESN

) evals

WHERE

(evals.NUMFLAGS / evals.ASPIRATIONS) >= 0.02

AND evals.ASPIRATIONS >= 10

**Apollo Algorithm Details**

(\* is Mandatory)

|  |  |
| --- | --- |
| **Apollo Details** |  |
| Algorithm ID \* | AlinityIA\_R1PipettorFEPressure |
| Algorithm Name \* | AI R1 FE Pressure |
| Algorithm Description \* | To detect degrading Pipettor Pressure Transducers through raised Front End (FE) pressure readings during aspiration. |
| Product Family \* | Alinity IA |
| Algorithm Group \* | Alinity Pipettor |
| Functional Area | N/A |
| Algorithm Category 1 | N/A |
| Algorithm Category 2 | N/A |
| Algorithm Category 3 | N/A |
| Remaining Useful Life Value | 3 |
| Remaining Useful Life Unit | Day |
| Keep Results Num Days | 14 |
| **Routine Details** |  |
| Routine Source | Use Reusable Routine |
| Reusable Routines | PHM\_ICQ\_Pipettor\_FE\_PROC |
| Routine Type | Oracle Procedure |
| Run Mode | Batch |
| Routine Invoke Command | PHM\_ICQ\_Pipettor\_FE\_PROC |
| Routine Procedure (SQL file) | PHM\_ICQ\_Pipettor\_FE\_PROC.sql |
| Status | Enable |
| Repeat Deploy | NO |
| **ODS Routine Details** |  |
| ODS Routine Name | PHM\_ODS\_ICQ\_PMEVENTS\_PROC |
| **Predictive Health Notification Details** |  |
| PHN Code | CC11 |
| PHN Level 1 Description | PHN\_Alinity\_IA |
| PHN Level 2 Description | PHN\_R1 Pipettor\_IA |
| PHN Level 3 Description | FE Pressure |
| **Knowledge Management DB Articles** |  |
| KM Article ID | K82525012 |
| KM Article | PHN\_IA: Front End (FE) Pressure |
| KM Link | <https://addkm.abbott.com/advisor/showcase?case=K82525012&project=Abbott> |
| **Parameters** |  |
| Parameter Group Name | FEPRESSURE\_THRESHOLD |
| Parameter Name | I\_FEPRESSURE\_THRESHOLD\_PIPMECHNAME,  I\_FEPRESSURE\_THRESHOLD\_ASPS,  I\_FEPRESSURE\_THRESHOLD\_MIN, I\_FEPRESSURE\_THRESHOLD\_MAX,  I\_FEPRESSURE\_THRESHOLD\_PCTASPS |
| Parameter Values | “Reagent1PipettorMechanism”, 10, 21000, 27000, 0.02 |
| **Chart Details** |  |
| Chart Title | AI R1 FE Pressure |
| Chart Type | Line |
| Chart Threshold Parameter | Threshold |
| Group ID | Group 7 |
| Chart X Axis Name | Date Time |
| Chart Y Axis Name | Pressure |

**APPENDIX 2:** CC21 PHN\_Alinity\_IA; PHN\_R2 Pipettor; FE Pressure

**Algorithm Code**

SELECT

evals.MODULESN

FROM

(SELECT

PM.MODULESN,

COUNT(PM.PIPETTORMECHANISMNAME) AS ASPIRATIONS,

SUM(CASE WHEN PM.FRONTENDPRESSURE > 27000 OR PM.FRONTENDPRESSURE < 21000

THEN 1

ELSE 0

END) AS NUMFLAGS

FROM

IDAQOWNER.ICQ\_PMEVENTS PM

WHERE

PM.LOGDATE\_LOCAL >= TRUNC(SYSDATE) - 1

AND PM.LOGDATE\_LOCAL < TRUNC(SYSDATE)

AND PM.FRONTENDPRESSURE IS NOT NULL

AND PM.PIPETTINGPROTOCOLNAME != 'NonPipettingProtocol'

AND PM.PIPETTORMECHANISMNAME = 'Reagent2PipettorMechanism'

GROUP BY

PM.MODULESN

ORDER BY

PM.MODULESN

) evals

WHERE

(evals.NUMFLAGS / evals.ASPIRATIONS) >= 0.02

AND evals.ASPIRATIONS >= 10

**Apollo Algorithm Details**

(\* is Mandatory)

|  |  |
| --- | --- |
| **Apollo Details** |  |
| Algorithm ID \* | AlinityIA\_R2PipettorFEPressure |
| Algorithm Name \* | AI R2 FE Pressure |
| Algorithm Description \* | To detect degrading Pipetter Pressure Transducers through raised Front End (FE) pressure readings during aspiration. |
| Product Family \* | Alinity IA |
| Algorithm Group \* | Alinity Pipettor |
| Functional Area | N/A |
| Algorithm Category 1 | N/A |
| Algorithm Category 2 | N/A |
| Algorithm Category 3 | N/A |
| Remaining Useful Life Value | 3 |
| Remaining Useful Life Unit | Day |
| Keep Results Num Days | 14 |
| **Routine Details** |  |
| Routine Source | Use Reusable Routine |
| Reusable Routines | PHM\_ICQ\_Pipettor\_FE\_PROC |
| Routine Type | Oracle Procedure |
| Run Mode | Batch |
| Routine Invoke Command | PHM\_ICQ\_Pipettor\_FE\_PROC |
| Routine Procedure (SQL file) | PHM\_ICQ\_Pipettor\_FE\_PROC.sql |
| Status | Enable |
| Repeat Deploy | NO |
| **ODS Routine Details** |  |
| ODS Routine Name | PHM\_ODS\_ICQ\_PMEVENTS\_PROC |
| **Predictive Health Notification Details** |  |
| PHN Code | CC21 |
| PHN Level 1 Description | PHN\_Alinity\_IA |
| PHN Level 2 Description | PHN\_R2 Pipettor\_IA |
| PHN Level 3 Description | FE Pressure |
| **Knowledge Management DB Articles** |  |
| KM Article ID | K82525012 |
| KM Article | PHN\_IA: Front End (FE) Pressure |
| KM Link | <https://addkm.abbott.com/advisor/showcase?case=K82525012&project=Abbott> |
| **Parameters** |  |
| Parameter Group Name | FEPRESSURE\_THRESHOLD |
| Parameter Name | I\_FEPRESSURE\_THRESHOLD\_PIPMECHNAME,  I\_FEPRESSURE\_THRESHOLD\_ASPS,  I\_FEPRESSURE\_THRESHOLD\_MIN, I\_FEPRESSURE\_THRESHOLD\_MAX,  I\_FEPRESSURE\_THRESHOLD\_PCTASPS |
| Parameter Values | “Reagent2PipettorMechanism”, 10, 21000, 27000, 0.02 |
| **Chart Details** |  |
| Chart Title | AI R1 FE Pressure |
| Chart Type | Line |
| Chart Threshold Parameter | Threshold |
| Group ID | Group 7 |
| Chart X Axis Name | Date Time |
| Chart Y Axis Name | Threshold |

**APPENDIX 3:** CCJ1 PHN\_Alinity\_IA; PHN\_Sample Pipettor; FE Pressure

**Algorithm Code**

SELECT

evals.MODULESN

FROM

(SELECT

PM.MODULESN,

COUNT(PM.PIPETTORMECHANISMNAME) AS ASPIRATIONS,

SUM(CASE WHEN PM.FRONTENDPRESSURE > 27000 OR PM.FRONTENDPRESSURE < 21000

THEN 1

ELSE 0

END) AS NUMFLAGS

FROM

IDAQOWNER.ICQ\_PMEVENTS PM

WHERE

PM.LOGDATE\_LOCAL >= TRUNC(SYSDATE) - 1

AND PM.LOGDATE\_LOCAL < TRUNC(SYSDATE)

AND PM.FRONTENDPRESSURE IS NOT NULL

AND PM.PIPETTINGPROTOCOLNAME != 'NonPipettingProtocol'

AND PM.PIPETTORMECHANISMNAME = 'SamplePipettorMechanism'

GROUP BY

PM.MODULESN

ORDER BY

PM.MODULESN

) evals

WHERE

(evals.NUMFLAGS / evals.ASPIRATIONS) >= 0.02

AND evals.ASPIRATIONS >= 10

**Apollo Algorithm Details**

(\* is Mandatory)

|  |  |
| --- | --- |
| **Apollo Details** |  |
| Algorithm ID \* | AlinityIA\_SamplePipettorFEPressure |
| Algorithm Name \* | AI S FE Pressure |
| Algorithm Description \* | To detect degrading Pipetter Pressure Transducers through raised Front End (FE) pressure readings during aspiration. |
| Product Family \* | Alinity IA |
| Algorithm Group \* | Alinity Pipettor |
| Functional Area | N/A |
| Algorithm Category 1 | N/A |
| Algorithm Category 2 | N/A |
| Algorithm Category 3 | N/A |
| Remaining Useful Life Value | 3 |
| Remaining Useful Life Unit | Day |
| Keep Results Num Days | 14 |
| **Routine Details** |  |
| Routine Source | Use Reusable Routine |
| Reusable Routines | PHM\_ICQ\_Pipettor\_FE\_PROC |
| Routine Type | Oracle Procedure |
| Run Mode | Batch |
| Routine Invoke Command | PHM\_ICQ\_Pipettor\_FE\_PROC |
| Routine Procedure (SQL file) | PHM\_ICQ\_Pipettor\_FE\_PROC.SQL |
| Status | Enable |
| Repeat Deploy | NO |
| **ODS Routine Details** |  |
| ODS Routine Name | PHM\_ODS\_ICQ\_PMEVENTS\_PROC |
| **Predictive Health Notification Details** |  |
| PHN Code | CCJ1 |
| PHN Level 1 Description | PHN\_Alinity\_IA |
| PHN Level 2 Description | PHN\_Sample Pipettor\_IA |
| PHN Level 3 Description | FE Pressure |
| **Knowledge Management DB Articles** |  |
| KM Article ID | K82525012 |
| KM Article | PHN\_IA: Front End (FE) Pressure |
| KM Link | <https://addkm.abbott.com/advisor/showcase?case=K82525012&project=Abbott> |
| **Parameters** |  |
| Parameter Group Name | FEPRESSURE\_THRESHOLD |
| Parameter Name | I\_FEPRESSURE\_THRESHOLD\_PIPMECHNAME,  I\_FEPRESSURE\_THRESHOLD\_ASPS,  I\_FEPRESSURE\_THRESHOLD\_MIN, I\_FEPRESSURE\_THRESHOLD\_MAX,  I\_FEPRESSURE\_THRESHOLD\_PCTASPS |
| Parameter Values | SamplePipettorMechanism”, 10, 21000, 27000, 0.02 |
| **Chart Details** |  |
| Chart Title | AI S FE Pressure |
| Chart Type | Line |
| Chart Threshold Parameter | Threshold Count |
| Group ID | Group 7 |
| Chart X Axis Name | Date Time |
| Chart Y Axis Name | Threshold |

**APPENDIX 4:** Algorithm Understanding Check – Algorithm Developer to Predictive Health Monitoring (PHM) Specialist Transition

**Data Set Description**

The data set for this understanding check was retrieved from the ICQOWNER.ODR\_PMEVENTSICQ table within the BSQD1I database. Data was collected for all instruments between December, 1 2016 and February 28, 2017, inclusive. Data collection was limited to the SERIALNUMBER, PIPETTORMECHANISMNAME, FRONTEDNPRESSURE, PIPETTINGPROTOCOLNAME, and DATETIMESTAMP fields.

**Data Set Retrieval**

The following SQL code was used to retrieve the data set:

SELECT

PM.SERIALNUMBER,

PM.PIPETTORMECHANISMNAME,

PM.FRONTENDPRESSURE,

PM.PIPETTINGPROTOCOLNAME,

PM.DATETIMESTAMP

FROM

ICQOWNER.ODR\_PMEVENTSICQ PM

WHERE

TRUNC(PM.DATETIMESTAMP) >= TO\_DATE('12/01/2016 12:00:00 AM', 'mm/dd/yyyy hh:mi:ss am')

AND TRUNC(PM.DATETIMESTAMP) < TO\_DATE('03/01/2017 12:00:00 AM', 'mm/dd/yyyy hh:mi:ss am')

**Algorithm Developer Analysis**

The following JMP script (JMP version 12.1.0) was used by the Algorithm Developer to analyze the data set and flag algorithm violations:

clear log**()**;

include**(**"JMPCreds\_v1.jsl"**)**;

// Variables

// \_\_user\_name and \_\_user\_pw are utilized by the GetCreds function within JMPCreds\_v1.jsl

time\_start = Informat**(** "12/01/2016", "mm/dd/yyyy" **)**;

time\_end = Informat**(** "03/01/2017", "mm/dd/yyyy" **)**;

// Database Credentials and Connectino to Database

GetCreds;

dbc = Create Database Connection**(**

"DSN=BSQD1i;UID=" || \_\_user\_name || ";PWD=" || Hex To Char**(** \_\_user\_pw **)** || ";

DBQ=BSQD1I; DBA=W;APA=T;EXC=F;FEN=T;QTO=T;FRC=10;FDL=10;LOB=T;RST=T;BTD=F;BNF=F;

BAM=IfAllSuccessful;NUM=NLS;DPM=F;MTS=T;MDI=F;CSR=F;FWC=F;FBS=1000000000000;TLO=O;MLD=0;ODA=F;"

**)**;

// SQL Queries

dt = Execute SQL**(** dbc,

"SELECT

t1.MODULESNDRM, t1.SERIALNUMBER, t1.DATETIMESTAMP, t1.PIPETTINGPROTOCOL,

t1.PIPETTINGPROTOCOLNAME, t1.PIPETTORMECHANISM, t1.PIPETTORMECHANISMNAME,

t1.FRONTENDPRESSURE

FROM

ICQOWNER.ODR\_PMEVENTSICQ t1

WHERE

( ( ( t1.DATETIMESTAMP >= TO\_TIMESTAMP('" || char**(** time\_start **)** || "', 'DDMONYYYY:HH24:MI:SS') ) AND

( t1.DATETIMESTAMP < TO\_TIMESTAMP('" || char**(** time\_end **)** || ":00:00:00', 'DDMONYYYY:HH24:MI:SS') ) ) ) ;",

char**(**time\_start**)** || "to" || char**(**time\_end**)** || "\_ICQ\_FETable" **)**;

// Close Connection

Close Database Connection**(**dbc**)**;

dt << **select where(**

dt:PIPETTINGPROTOCOL != **8**

**)**;

dt\_sub = dt << **Subset(**

**(** Selected Rows **)**,

Output Table Name**(**"Protocol\_Filtered"**)**

**)**;

dt\_sub << **new column(**

"Date",

numeric,

nominal,

Format**(** "m/d/y" **)**,

Formula**(**

num**(** Short Date**(** dt\_sub:DATETIMESTAMP **)** **)**

**)**

**)**;

dt\_sum = dt\_sub << **Summary(**

Group**(** dt\_sub:SERIALNUMBER, dt\_sub:PIPETTORMECHANISMNAME, dt\_sub:Date **)**,

Min**(** dt\_sub:FRONTENDPRESSURE **)**,

Max**(** dt\_sub:FRONTENDPRESSURE **)**,

Freq**(** "None" **)**,

Weight**(** "None" **)**

**)**;

dt\_sum << **new column(**

"Flag\_Status",

character,

continuous

**)**;

for**(** i = **1**, i <= NRow**(** dt\_sum **)**, i++,

if**(** Column**(** dt\_sum, "N Rows" **)[**i**]** < **10**,

dt\_sum:Flag\_Status**[**i**]** = "LessThan10",

if**(** Column**(** dt\_sum, "Min(FRONTENDPRESSURE)" **)[**i**]** >= **21000** &

Column**(** dt\_sum, "MAX(FRONTENDPRESSURE)" **)[**i**]** <= **27000**,

dt\_sum:Flag\_Status**[**i**]** = "good",

dt\_sum:Flag\_Status**[**i**]** = "flagged"

**)**

**)**

**)**;

dt\_sum << **Sort(**

Replace Table,

By**(** :Flag\_Status**)**,

Order**(** Ascending **)**

**)**;

**PHM Specialist Analysis**

The following SQL code was used by the PHM Specialist to analyze the data set and flag algorithm violations:

SELECT

evals.SERIALNUMBER,

evals.PIPETTORMECHANISMNAME,

evals.DAY,

evals.MAXFEP,

evals.MINFEP

FROM

(SELECT

PM.SERIALNUMBER,

PM.PIPETTORMECHANISMNAME,

TRUNC(PM.DATETIMESTAMP) AS DAY,

MAX(PM.FRONTENDPRESSURE) AS MAXFEP,

MIN(PM.FRONTENDPRESSURE) AS MINFEP,

COUNT(PM.PIPETTORMECHANISMNAME) AS ASPIRATIONS,

SUM(CASE WHEN PM.FRONTENDPRESSURE > 27000 OR PM.FRONTENDPRESSURE < 21000

THEN 1

ELSE 0

END) AS NUMFLAGS

FROM

ICQOWNER.ODR\_PMEVENTSICQ PM

WHERE

TRUNC(PM.DATETIMESTAMP) >= TO\_DATE('12/01/2016 12:00:00 AM', 'mm/dd/yyyy hh:mi:ss am')

AND TRUNC(PM.DATETIMESTAMP) < TO\_DATE('03/01/2017 12:00:00 AM', 'mm/dd/yyyy hh:mi:ss am')

AND PM.FRONTENDPRESSURE IS NOT NULL

AND PM.PIPETTINGPROTOCOLNAME != 'NonPipettingProtocol'

GROUP BY

PM.SERIALNUMBER,

PM.PIPETTORMECHANISMNAME,

TRUNC(PM.DATETIMESTAMP)

ORDER BY

PM.SERIALNUMBER,

TRUNC(PM.DATETIMESTAMP)

) evals

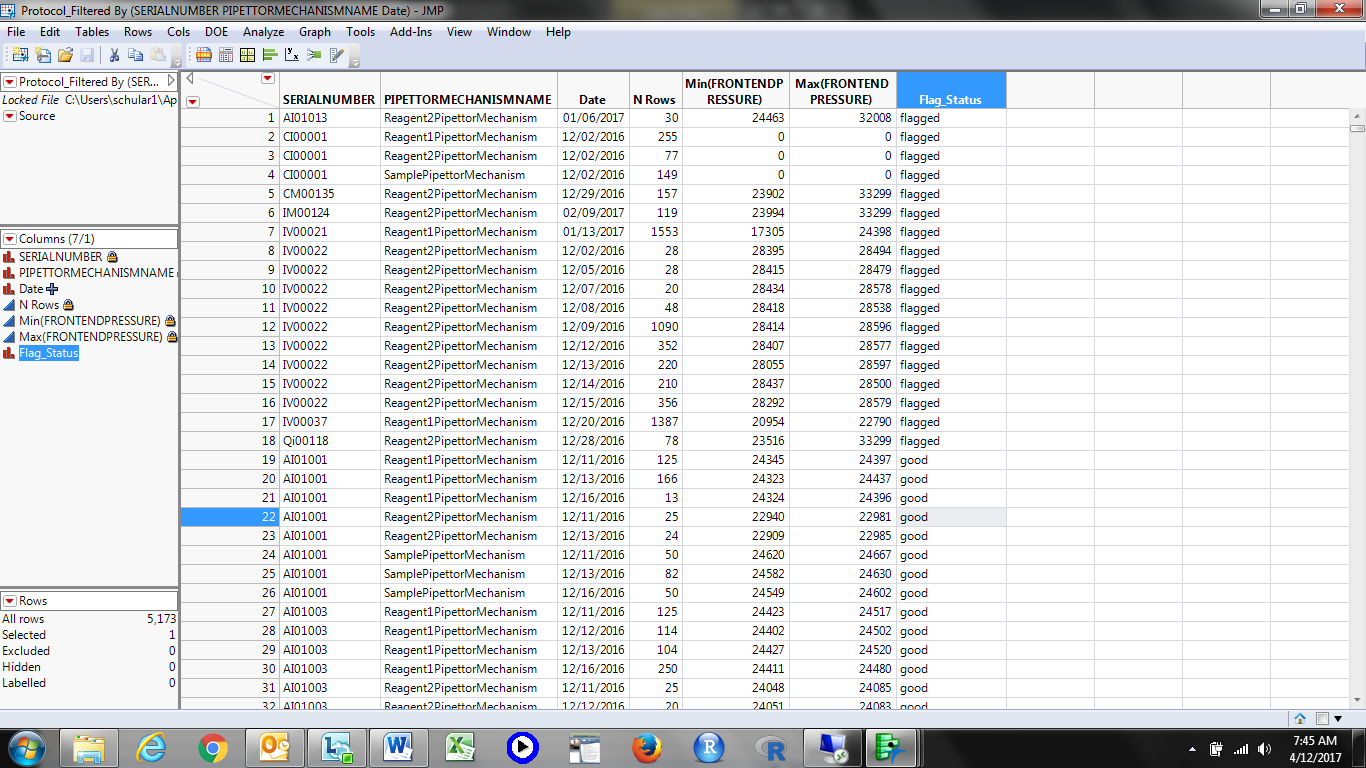
WHERE

evals.NUMFLAGS >= 1

AND evals.ASPIRATIONS >= 10

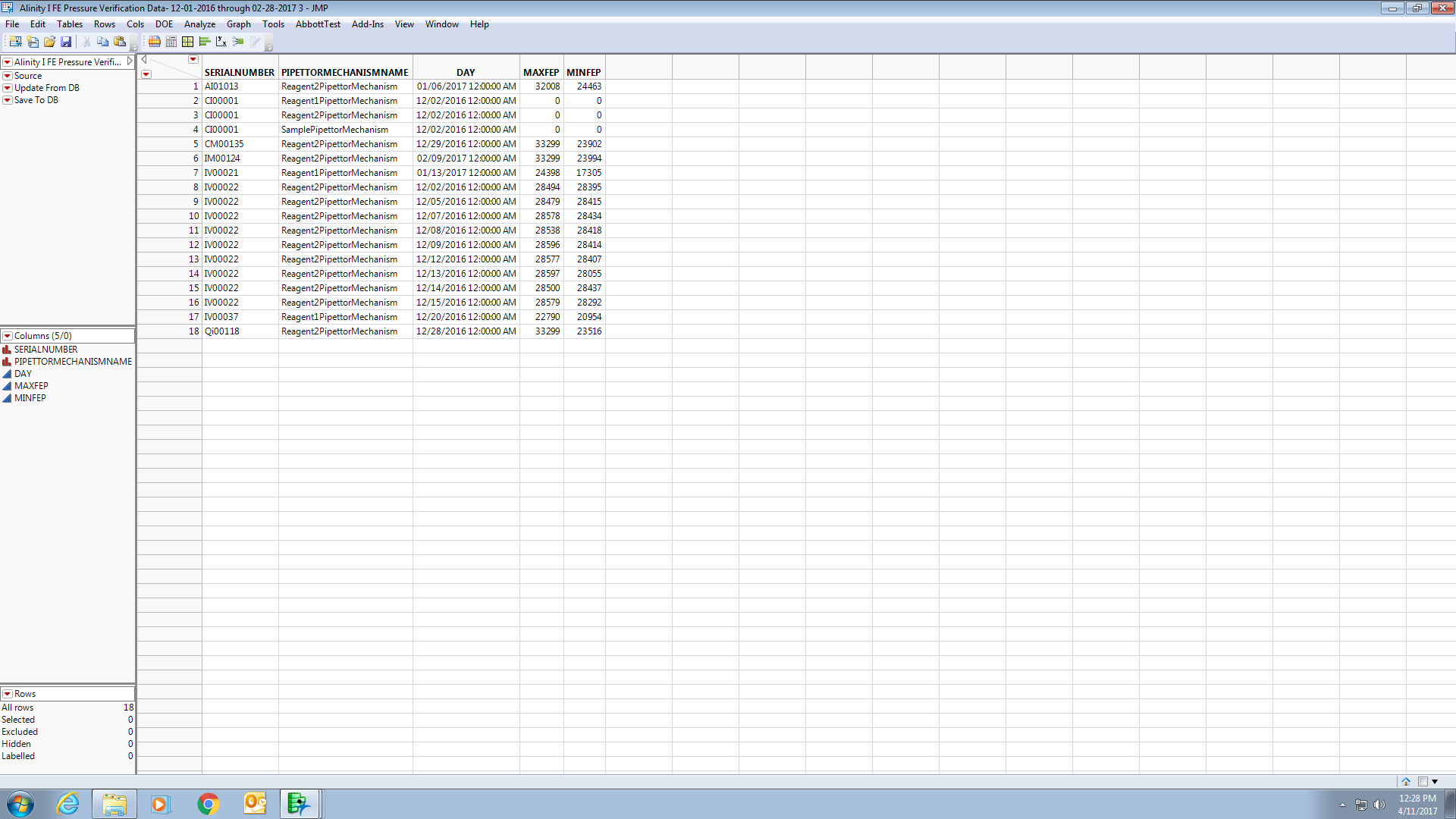
**Algorithm Developer Analysis Output**

The following 18 instrument (SERIALNUMBER) and pipettor (PIPETTORMECHANISMNAME) combinations were identified as violating the algorithm by the Algorithm Developer:



**PHM Specialist Analysis Output**

The following 18 instrument (SERIALNUMBER) and pipettor (PIPETTORMECHANISMNAME) combinations were identified as violating the algorithm by the PHM Specialist:



**Algorithm Developer & PHM Specialist Output Comparison**

|  |  |  |  |
| --- | --- | --- | --- |
| Total # of Unique Instrument-Part-Days Tested | Total # of Algorithm Developer Flags | Total # of PHM Specialist Flags | Total # of Matched Flags (Algorithm Developer vs. PHM Specialist) |
| **5,874** | **18** | **18** | **18** |

**Understanding Check Summary**

Based on the outputs from both the Algorithm Developer and PHM Specialist, the PHM Specialist’s understanding of the delivered algorithm is confirmed. Both the Algorithm Developer and PHN Specialist analyzed the same data set and got the same results. In particular, the SERIALNUMBER, PIPETTORMECHANISMNAME, and Date/DAY fields matched for all 18 instrument-part-day (SERIALNUMBER-PIPETTORMECHANISMNAME-Date/DAY) combinations. This means that both the Algorithm Developer and PHM Specialist flagged the same 18 algorithm violations within the given data set. Furthermore, there were no mismatches between the output from the Algorithm Developer and the output from the PHM Specialist.

Note: Although not explicitly called out in the understanding check information above, as it was not part of the algorithm originally delivered by the Algorithm Developer, the additional 2% requirement to check for occluded probes was implemented, tested, and behavior confirmed by the PHM Specialist. Moreover, when applied to the understanding check data set described above, the 2% requirement reduced the number of algorithm flags to 16. Two instrument-part-day combinations had only 1flagged aspiration out of many; a proportion less than 2%.

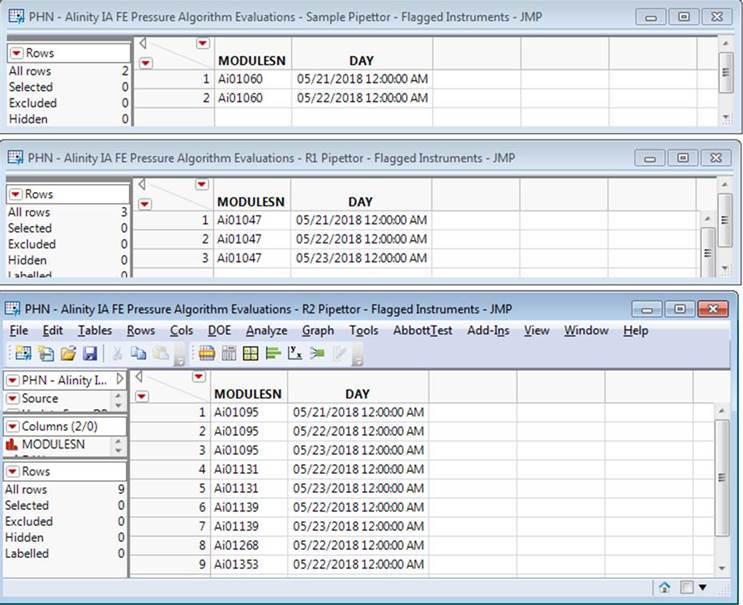
**APPENDIX 5:** Algorithm Transition to Apollo – PHM Specialist to Apollo Developer

**Data Set Description**

The data set for this transition was retrieved from the IDAQOWNER.ICQ\_PMEVENTS table within the DABBTO database. Data was collected for all available instruments between May, 21 2018 and May 23, 2018, inclusive.

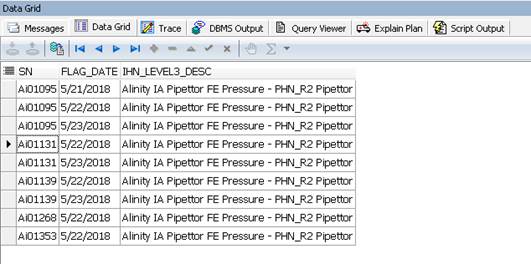
**PHM Specialist Analysis Output**

The following 14 instrument-days (MODULESN-DAY) were identified as violating the algorithm by the PHM Specialist:



**Apollo Developer Analysis Output**

The following 9 instrument-days (SN-FLAG\_DATE) were identified as violating the algorithm by the Apollo Developer (note that where the Apollo Developer and PHM Specialist don’t match is because the instruments do not exist in the Apollo environment, and therefore have no data to run the algorithm against):



**Algorithm Transition Summary**

Based on the outputs from both the Apollo Developer and PHM Specialist, the Apollo Developer’s understanding of the delivered algorithm is confirmed. Both the Apollo Developer and PHM Specialist analyzed the same data set and got the same results (with the exception of the instruments not in Apollo as mentioned previously). In particular, the MODULESN/SN and DAY/FLAG\_DATE fields matched. This means that both the Apollo Developer and PHM Specialist flagged the same algorithm violations within the given data set.