Gating-ML Compliance Test Suite

This test suite is intended to determine/verify Gating-ML compliance of third party analytical software in terms of:

(i) being able to read Gating-ML compliance files,
(ii) being able to apply gates on list mode data files,
(iii) being able to perform appropriate data transformations for gates described on transformed data, and
(iv) obtaining correct events membership results for described gates.

The test suite contains 32 sets of tests (Gating-ML files) totaling 461 different gates, appropriate list mode data files, and detailed expected event membership information for each of the gate. Each test set is intended to test a particular gate type, transformation type, or Gating-ML feature, as indicated by the name of the test set.

Files in this test suite are organized as follows:

- The `Summary.csv` file contains an overview of all tests/gates. Each line represents a gate. Columns provide the name of the test set, the identifier of the gate, the name of the Gating-ML file, the name of the corresponding list mode data file, and the number of events expected to be found inside the gate. The last column states whether the test has been identified as sensitive to the precision of floating point computing.

- The `Gating-ML Files` folder contains all Gating-ML files. Each of the files represents a single test set.

- The `List-mode Data Files` folder contains FCS data files (name.fcs) and their descriptions (name.pdf). List-mode data files are shared among several test sets.

- The `Expected Results` folder contains a subfolder for each of the test sets. The name of the subfolder corresponds to the name of the test set. Each of these folders contains several text-based result files – one for each gate. The name of the file corresponds to the name of the gate (with the .txt extension). First line in the text file repeats the gate identifier. An ordered sequence of 0/1 follows, one number per line. This sequence corresponds to the order of events appearing in the appropriate list mode data file. The value zero (0) indicates that the event is not in the gate, the value one (1) indicates that the event is in the gate.

  Please note that some of the files contain a single number. This is as some of the gates are intended to be tested against a data file containing a single event only.

  In addition, some of these folders contain a .csv file. This is included in test sets that include transformations. The .csv file provides a reference of the calculated transformed data.

This work is supported by NIH grant EB005034

Floating Point Precision Issues

The included expected results provide information for each gate and each appropriate event whether or not it ideally should be found in the gate. These are only informative values. Unfortunately, due to machine approximation of real numbers and limited precision of floating point arithmetic, different software may output different results for events *too close* to the boundary of a gate, or even the same software application may provide different results when running on a different hardware or operating system. While these differences are very tiny and not significant considering other sources of variation (biological, instruments, etc.); they may create confusion about whether a certain software application is Gating-ML compliant.

In order to eliminate or at least minimize these artifacts, we suggest software developers to use the *default numeric machine precision* (see below) to test for equalities and inequalities. The approach that we are suggesting is based on the book Object-oriented Implementation of Numerical Methods by Didier H. Besset (published by Morgan Kaufmann, 2001, ISBN 1558606793, 9781558606791). We define the following two constants:

- *Machine Precision* being the largest positive number that when added to 1.0 yields 0.
- *Default Numeric Machine Precision* as the square root of the *Machine Precision*.

For the purpose of deciding event membership, software should consider the numbers *a* and *b* as equal if the difference between *a* and *b* is less than the *Default Numeric Machine Precision*. Java Reference implementation calculating the *Machine Precision* and *Default Numeric Machine Precision* and implementing an appropriate equality test can be found at http://archive.devx.com/java/free/book.asp. The MACHAR routine (http://orion.math.iastate.edu/burkardt/c_src/machar/machar.html) can be used to compute the machine floating point constraints in C and Fortran. On our testing platform, the *Default Numeric Machine Precision* equals approximately 1.0537e-8 in Java and 1.4901e-8 in R.

Within this test suite, the following gates/events have been identified as sensitive to floating point computing precision due to their position on (or very close to) the gate boundary. This may not be a complete list, please report any additional precision sensitive gates/events that you may identify.

Test Set	FCS Data File	Gate	Precision Sensitive Events (Enumerated)
03LargeRectangular	int-homogenous_matrix.fcs	Gatep1	11, 36, 61, 86, …, 1211, 1236 (50 events with FS=440)
03LargeRectangular	int-homogenous_matrix.fcs	Gatep2	11, 36, 61, 86, …, 1211, 1236 (50 events with FS=440)
03LargeRectangular	int-homogenous_matrix.fcs	Gatep3	11, 36, 61, 86, …, 1211, 1236 (50 events with FS=440)
03LargeRectangular	int-homogenous_matrix.fcs	Gatep4	11, 36, 61, 86, …, 1211, 1236 (50 events with FS=440)
03LargeRectangular	int-homogenous_matrix.fcs	Gatep5	11, 36, 61, 86, …, 1211, 1236 (50 events with FS=440)
03LargeRectangular	int-homogenous_matrix.fcs	Gatep6	11, 36, 61, 86, …, 1211, 1236 (50 events with FS=440)
03LargeRectangular	int-homogenous_matrix.fcs	Gatep7	11, 36, 61, 86, …, 1211, 1236 (50 events with FS=440)
22TrSqrt	int-10_events_6_parameters.fcs	TrSqrtG6	5, 9
19TrCombination	int-10_events_8_parameters.fcs	TRComG10	4, 10

Revision History

This section summarizes significant changes since the first published version of the Gating-ML Test Suite version 1.5.080410:

Version 1.5.081030

- The following Gating-ML files have been corrected (affected gates and/or transformations enumerated in brackets): 09Ellipsoids.xml (Ellipsoid_14i, Ellipsoid_15i, Ellipsoid_17o, Ellipsoid_20o), 10PolytopeGates.xml (polytope_18i), 28TrEH.xml (TrEH_G6), 30TrInvSplitScale.xml (ISpSc_G1), 24TrExp.xml (TrExp_1, TrExp_2, TrExp_3, TrExp_4, TrExpG1, TrExpG2, TrExpG3, TrExpG4), 01Rectangular.xml (EqualToMinAndMax, MinGreaterThanMax – these gates have been removed from the tests since – if both the *min* attribute and the *max* attribute are present then the value of the *min* attribute shall be smaller than the value of the *max* attribute according to the current version of the Gating-ML specification).
- Ellipsoid_19i.txt has been added to the expected results (the file was missing in the previous distribution).
- The offset information in the text segment of the int-10_events_8_parameters.fcs has been corrected (the value of the $BEGINDATA did not match the offset information in the header segment)
- Detailed expected results have been updated for the comp2G5 gate (comp2G5.txt in 32TrCompensation2)
- Floating point precision issues have been addressed in the compliance tests readme.pdf file a “precision sensitive” column has been added to the Summary.csv file.

Version 1.5.081007

- Expected results for the following gates have been updated: prt_p08, prt_p09, prt_23, prt_08, prt_02, prt_19 and prt_03 (all these gates are in 17ParentIdTest.xml and they are related to rounding errors effecting events close to the boundary of an ellipsoid gate that is not "straight" with the coordinate system.
- This test suite is for Gating-ML version 1.5.081007 while the previous one was for Gating-ML 1.5.080311. Gating-ML 1.5.080311 was ambigous in the definiton of the *sinh* transformation (Eq. 23 did not correspond to definition in XML schema and documentation from MathML Central) and therefore tests using *sinh* could have been considered wrong depending on which definition of *sinh* has been used. Eq. 23 has been fixed in 1.5.081007, which implicitely “fixes” related tests without changes to the expected values of *sinh* related transformations and gates.
- Typos have been corrected in parameter names in the LComb_4 transformation within the 19TrCombination.xml file.
- A non-significant lowercase/uppercase typo has been corrected in a parameter name in the 17ParentIdTest.xml file.