Cytometry
PART A
Journal of the International Society for Advancement of Cytometry

Gating-ML: XML-Based Gating Descriptions in Flow Cytometry

Josef Spidlen,[1] Robert C. Leif,[2] Wayne Moore,[3] Mario Roederer,[4] International Society for the Advancement of Cytometry Data Standards Task Force, Ryan R. Brinkman[1]*

[1] Terry Fox Laboratory, BC Cancer Agency, Vancouver, British Columbia, Canada

[2] Newport Instruments, San Diego, California

[3] Genetics Department, Stanford University School of Medicine, Stanford, California

[4] NIH Vaccine Research Center, Bethesda, Maryland

Received 19 June 2008; Accepted 5 August 2008

Additional Supporting Information may be found in the online version of this article.

Conflict of interest: Nothing to report.

Grant sponsor: National Institute of Biomedical Imaging and Bioengineering; Grant number: EB005034.

*Correspondence to: Ryan R. Brinkman, Terry Fox Laboratory, BC Cancer Research Centre, 675 West 10th Avenue, Vancouver, BC V5Z 1L3, Canada

Email: rbrinkman@bccrc.ca

Published online 4 September 2008 in Wiley InterScience (www.interscience.wiley.com)

DOI: 10.1002/cyto.a.20637

International Society for Advancement of Cytometry

Abstract

The lack of software interoperability with respect to gating due to lack of a standardized mechanism for data exchange has traditionally been a bottleneck, preventing reproducibility of flow cytometry (FCM) data analysis and the usage of multiple analytical tools. To facilitate interoperability among FCM data analysis tools, members of the International Society for the Advancement of Cytometry (ISAC) Data Standards Task Force (DSTF) have developed an XML-based mechanism to formally describe gates (Gating-ML). Gating-ML, an open specification for encoding gating, data transformations and compensation, has been adopted by the ISAC DSTF as a Candidate Recommendation. Gating-ML can facilitate exchange of gating descriptions the same way that FCS facilitated for exchange of raw FCM data. Its adoption will open new collaborative opportunities as well as possibilities for advanced analyses and methods development. The ISAC DSTF is satisfied that the standard addresses the requirements for a gating exchange standard. © 2008 International Society for Advancement of Cytometry

Key terms

flow cytometry; gating; XML; data standard; compensation; transformation; bioinformatics

FLOW cytometry has traditionally been a manually intensive technique. However, automated high throughput FCM methods have recently been developed to rapidly collect large data sets with complexities similar to gene microarrays (1). The large amount of complex information generated by high throughput technologies needs to be processed in an automated, well describable way. One of the most insidious problems in accomplishing this goal is the lack of standard data formats for information exchange (2). A basic challenge for FCM is to simplify, from the end user's viewpoint, data analysis and extraction of statistical information (3–5).

Flow Cytometry Data File Standard

In 1984, the first FCM data file format (FCS) was adopted, followed by FCS 2.0 (6) in 1990, and the current FCS 3.0 (7) in 1997. This standard is supported by all analytical instrument and third party software suppliers with no major compatibility issues for the raw fluorescence values that FCS captures. The main component of a list mode FCS data set is a matrix of parameter values with "columns" corresponding to parameters and "rows" corresponding to electronic events. Additional information about the content of a data set is mostly specified through predefined keywords.

Gating Description in FCS

Within FCS files, Boolean collection of regions (defined by the $RnI and $RnW keywords) can be encoded by the $GATING keyword. A region can either represent a one-dimensional range gate or a two-dimensional polygon gate created in an untransformed data space. The $GATING keyword is intended to capture the gating conditions used for data acquisition and is not designed for gating performed as part of a post-acquisition analytical process.

Compensation Description in FCS

Parameter (fluorescence) compensation descriptions can be encoded in FCS for electronic compensation during data acquisition. However, as there may be several ways to compensate the data, it is preferred to store uncompensated values and describe (suggest) compensation for post-acquisition data processing. Unfortunately, using FCS to store this type of compensation may introduce confusion whether the data has been compensated or not. FCS 2.0 (6) includes a standardized set of $DFC*i*TO*j* keywords to store the percentage of the value of parameter *i* that shall be subtracted from the value of parameter *j* in order to compensate *j*. In FCS 3.0 (7), these keywords have been replaced with a single $COMP keyword storing the entire compensation matrix. Although there is little meaning behind a fragment of a compensation matrix, a fragment of a spillover matrix can still encode the spectral overlap of involved parameters. This added functionality has resulted in the $SPILLOVER keyword being proposed to replace $COMP in FCS 3.1 (8).

Data Transformations in FCS

The characteristics of FCM data and historical availability of log amplifiers lead to the support of logarithmic transformation in FCS. Based on the $P*n*R, $P*n*E, and $P*n*G keywords, the FCS standard allows the specification of measured values that have been amplified either logarithmically or linearly prior storing in FCS files. Although this is absolutely essential for correct visualization as well as for further post-acquisition transformations, it is insufficient for capturing of post-acquisition analytical steps.

Motivation for a Gating Description Standard

Gating descriptions cannot be adequately captured in FCS files. The lack of standardization limits collaboration, independent validation/refutation, and clinical validation and minimizes the value of the wealth of existing FCM data due to poor annotation (9). The lack of software interoperability on the gating level is a bottleneck, preventing independent reproducibility of FCM data analysis, usage of multiple analytical tools, and development of novel analytical and clinical methods. The need to describe gates has been recognized by the community, and developers of FCM analytical software tools commonly incorporate gating description in their proprietary files, such as workspace files, project files, or project template files. Unfortunately, these file formats include GUI-driven settings, are vendor-specific, closed-source, and incompatible among different vendors.

In this article, we present a solution with the potential of interconnecting analytical software to communicate details necessary to interpret FCM analysis. It gives the researcher the freedom of using any standard-compliant analytical software tool to exchange their analysis results with users of any other tool, the option of combining multiple tools together, as well a providing an open standard for documenting analysis methodologies.

Materials and Methods

For the past 2 years, members of the Flow Informatics and Computational Cytometry Society and the International Society for the Advancement of Cytometry (ISAC) Data Standards Task Force (DSTF) have been collaborating on development of new data standards for FCM (10). The general strategy has been to define the information required to be in the data file, prepare a requirements document to focus the development process, and finally to apply the previously mentioned to guide the development and review of new data standards.

This strategy has resulted in the MIFlowCyt criteria (11), and several data standard proposals including Gating-ML ("Gating Markup Language"), an open specification allowing for encoding of gating, data transformations and compensation. Gating-ML files are Extensible Markup Language (XML) (12) files describing gates in a way that is computationally reproducible and independent of any particular analytical software application. The Gating-ML specification has been developed reusing the methodology and best practices from international standardization bodies such as the World Wide Web Consortium (W3C), the Institute of Electrical and Electronics Engineers (IEEE), and the Internet Engineering Task Force (IETF). The design of Gating-ML XML schemas (13) is compliant with CytometryML (14,15).

Results

Gating-ML has undergone several revisions since the first public release in February 2006. At this point, the ISAC DSTF is satisfied that the specification serves its purpose, and has released it as a Candidate Recommendation (CR) to the ISAC membership and other interested parties. The Gating-ML specification is included as Supporting Information to this manuscript. Alternatively, it can be obtained directly from ISAC (16,17).

Components of the Standard

Gating-ML specifications consist of several components divided into normative and informative parts. The normative parts are crucial for a compliant implementation. They consist of a detailed description of Gating-ML and of XML schemas (13) defining the syntax of compliant files. In addition, several informative parts are included. These include example Gating-ML XML files, and HTML- and UML-based (18) documentation of the XML schemas. Informative compliance tests can also be downloaded (19). These tests include both example FCS files and Gating-ML files, along with the expected results of membership for all the types of gates included in the standard. It is expected that these example files will significantly aid the development and testing of software that implements the standard.

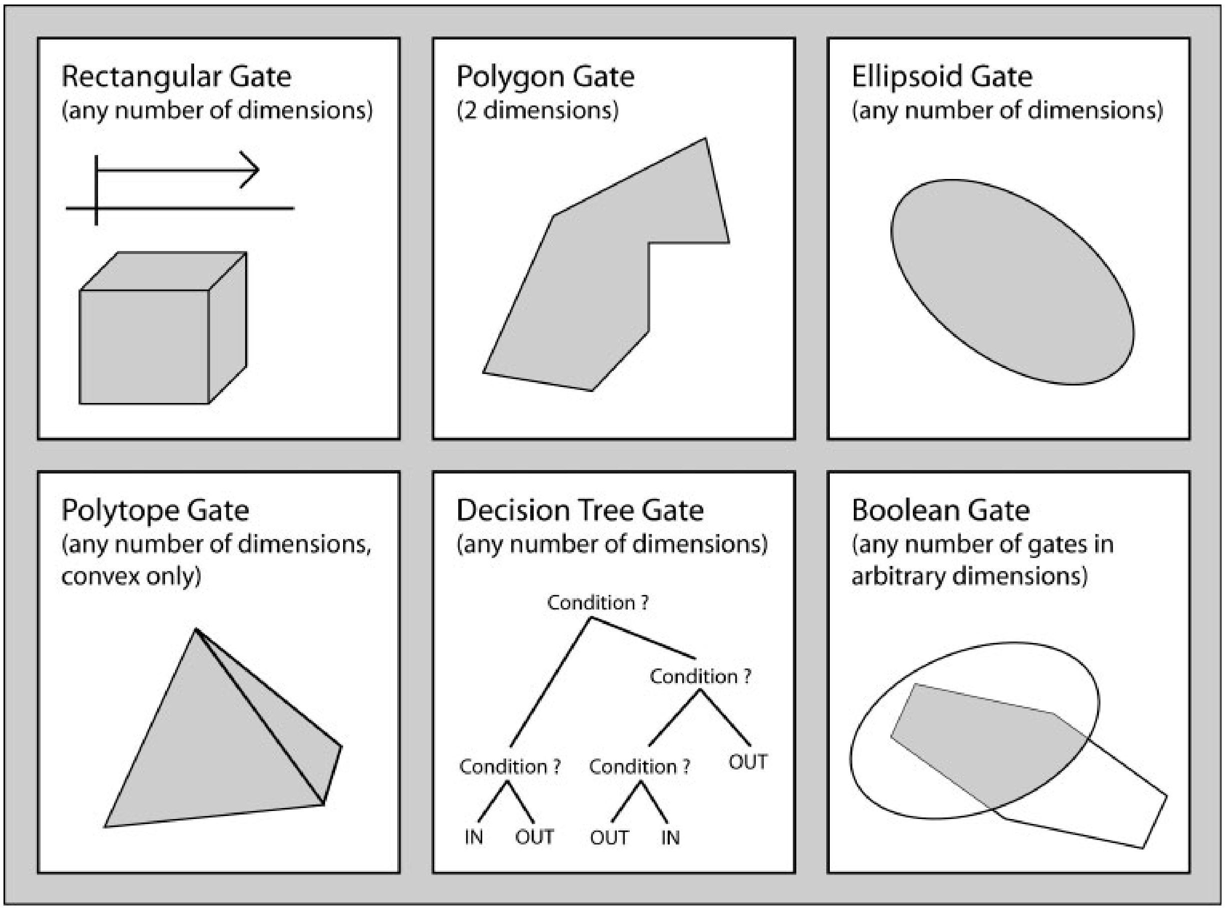

Rectangular Gate
(any number of dimensions)
Polygon Gate
(2 dimensions)
Ellipsoid Gate
(any number of dimensions)
Polytope Gate
(any number of dimensions, convex only)
Decision Tree Gate
(any number of dimensions)
Condition ?
Condition ?
Condition ?
Condition ?
OUT
IN
OUT
OUT
IN
Boolean Gate
(any number of gates in arbitrary dimensions)

Figure 1. Gate types supported by Gating-ML. Overview of the six types of gates supported by Gating-ML: rectangular gates in an arbitrary number of dimensions, i.e. going from one-dimensional range gates up to multidimensional hypercube gates; polygon gates, ellipsoid and hyper-ellipsoid gates, polytope gates, decision tree structures, and Boolean collections of any of the types of gates.

Scope and Features of the Standard

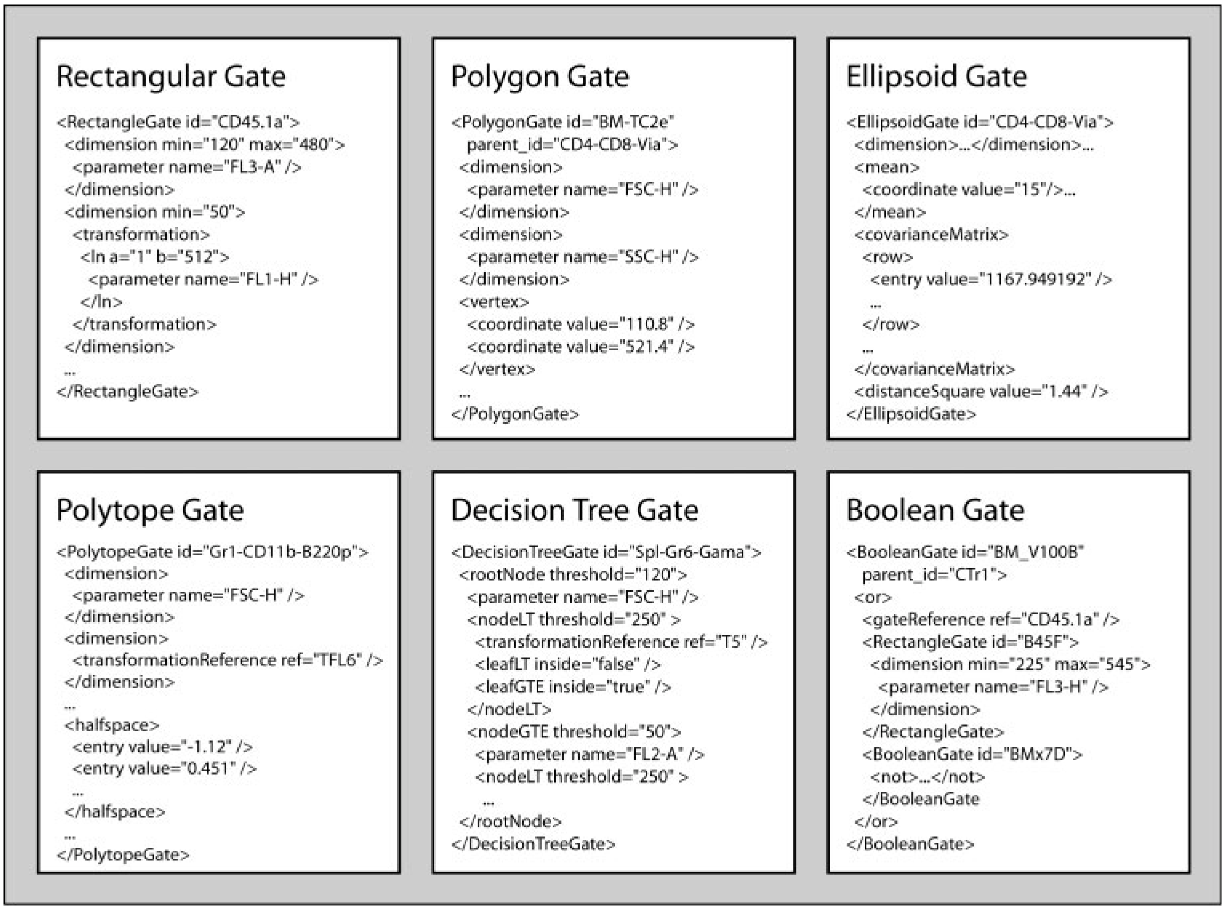
Gating-ML is an XML-based specification on how to form unambiguous gate definitions that are transferable between different software packages. It is a file format primarily serving the purpose of computationally exchanging details about post-acquisition analysis, and is not intended to define directly data acquisition or physical sorting gates. Gating-ML does not cover guidelines (protocols, standard operating procedures) specifying how gates shall be formed or what gating strategies shall be used for particular assays. However, in Gating-ML, gates can be ordered hierarchically so that each gate can either be applicable on the whole event population (as in a list mode data file), or on a subpopulation defined by another gate. Therefore, an arbitrary gating strategy may be encoded.

FCM data are commonly visualized on scales that do not directly correspond to the actual values stored in list mode data files. These user-friendly visualizations (20–22) can improve and simplify the interpretation of the data and analysis. If analyses involve transformed data, then the transformation needs to be described in order to reconstruct the analyses (11). In general, gates created in transformed space can effectively only be described in the transformed space. As transformations may not be reversible, and since nonlinear transformations have significant effects on the shape of gate boundaries, description of these gates in the data space would be both inaccurate and inefficient (see Supporting Information). Therefore, Gating-ML supports applying gates on both raw and transformed data. Either parameters from list mode data files directly, or transformations applied on these parameters create dimensions of the space where gates are defined. Transformations (which include fluorescence compensation) may also be combined into composite transformations.

Supported Gate Types

Gating-ML supports the following types of gates: rectangular, polygon, convex polytopes, ellipsoids, decision trees, and Boolean collections of any of the other types of gates. The six types of gates supported by Gating-ML are illustrated in Figure 1. XML-based definitions are demonstrated in the examples shown in Figure 2.

The most basic type of gates supported in Gating-ML are Rectangular gates in any number of dimensions, from



Rectangular Gate

```
<RectangleGate id="CD45.1a">
 <dimension min="120" max="480">
  <parameter name="FL3-A" />
 </dimension>
 <dimension min="50">
  <transformation>
   <ln a="1" b="512">
    <parameter name="FL1-H" />
   </ln>
  </transformation>
 </dimension>
 ...
</RectangleGate>
```

Polygon Gate

```
<PolygonGate id="BM-TC2e"
  parent_id="CD4-CD8-Via">
 <dimension>
  <parameter name="FSC-H" />
 </dimension>
 <dimension>
  <parameter name="SSC-H" />
 </dimension>
 <vertex>
  <coordinate value="110.8" />
  <coordinate value="521.4" />
 </vertex>
 ...
</PolygonGate>
```

Ellipsoid Gate

```
<EllipsoidGate id="CD4-CD8-Via">
 <dimension>...</dimension>...
 <mean>
  <coordinate value="15"/>...
 </mean>
 <covarianceMatrix>
  <row>
   <entry value="1167.949192" />
   ...
  </row>
  ...
 </covarianceMatrix>
 <distanceSquare value="1.44" />
</EllipsoidGate>
```

Polytope Gate

```
<PolytopeGate id="Gr1-CD11b-B220p">
 <dimension>
  <parameter name="FSC-H" />
 </dimension>
 <dimension>
  <transformationReference ref="TFL6" />
 </dimension>
 ...
 <halfspace>
  <entry value="-1.12" />
  <entry value="0.451" />
  ...
 </halfspace>
 ...
</PolytopeGate>
```

Decision Tree Gate

```
<DecisionTreeGate id="Spl-Gr6-Gama">
 <rootNode threshold="120">
  <parameter name="FSC-H" />
  <nodeLT threshold="250" >
   <transformationReference ref="T5" />
   <leafLT inside="false" />
   <leafGTE inside="true" />
  </nodeLT>
  <nodeGTE threshold="50">
   <parameter name="FL2-A" />
   <nodeLT threshold="250" >
    ...
 </rootNode>
</DecisionTreeGate>
```

Boolean Gate

```
<BooleanGate id="BM_V100B"
  parent_id="CTr1">
 <or>
  <gateReference ref="CD45.1a" />
  <RectangleGate id="B45F">
   <dimension min="225" max="545">
    <parameter name="FL3-H" />
   </dimension>
  </RectangleGate>
  <BooleanGate id="BMx7D">
   <not>...</not>
  </BooleanGate
 </or>
</BooleanGate>
```

Figure 2. Gating-ML description of gates. Examples of XML fragments demonstrating description of each of the types of gates. The XML structure is defined by Gating-ML XML schema and few additional validity requirements as specified in the Gating-ML normative description. Because of space constrains, examples have been simplified in that neither full XML nor fully qualified names (distribution into separate namespaces) are shown. Please consult examples included as Supporting Information for distribution into separate namespaces for gating, transformation, and others.

one-dimensional range gates up to multidimensional hyper-rectangular regions. Rectangular gates are defined by a set of one or more dimensions with the minimum (inclusive) and/or maximum (exclusive) thresholds specified for each dimension. Either the minimum or the maximum threshold may be left out to specify a one-side open gate.

Polygon gates in two dimensions represent one of the most common gates used for traditional manual gating when users draw borders around populations of interest. These gates are specified by an ordered sequence of vertices; the polygon is created by straight line segments spanned between consecutive vertices and between the last and the first vertex in the sequence. The polygon gate is defined as the interior of the polygon for simple polygons (boundary including), and by the Crossing Number method (23) for non-simple polygons, respectively. This definition allows for concave polygons and polygons with intersecting segments.

Convex polytope gates represent an extension of polygon gates into an unlimited number of dimensions. Convex polytope gates are defined by an intersection of half-spaces; each half space defined by a linear inequality. The polytope gate is defined as $G(A,b) = \{x: Ax + b \leq 0\}$, where A is an $m \times n$ matrix, m being the number of bounding half-spaces and n being the number of dimensions of the polytope; b is an $m \times 1$ column vector, 0 is a 0 column vector, and the inequality shall be met for each row. The coefficients of each row of A and b correspond to the coefficients of the linear inequality defining the particular half-space. This representation is computationally fast and easy to process to determine gate membership. These gates are unlikely to be created manually; however, they may be created as data driven gates by software performing (semi)automated analysis based on processing more than two parameters at the same time. Restricting the polytope gates to the "convex only" reduces the complexity significantly, whereas it does not affect the expressiveness of the Gating-ML language as any polytope gate is describable as a Boolean collection of several convex polytope gates.

Ellipsoid gates in two or more dimensions are the fourth type of gate supported in Gating-ML. Although two-dimensional ellipse gates are commonly supported by traditional analytical software, general ellipsoid and hyper-ellipsoid gates represent a straightforward extension into multidimensional space. These types of gates are expected to be one of the most typical outputs of advanced automated data driven gating, such as based on multidimensional clustering and multivariate normal modeling of the data. Representation of the ellipsoid gates has been designed to naturally support for the statistically driven use cases and to be computationally inexpensive to process. Therefore, the ellipsoid gates are defined by a covariance matrix, a mean vector, and a Mahalanobis distance (24). Specifically, the ellipsoid gate is defined as $G(\mu, C, D^2) = \{x: (x - \mu)^T C^{-1}(x - \mu) \leq D^2\}$, where μ is a column vector specifying the center of the ellipsoid, C^{-1} is the inverse of a covariance matrix, D^2 is the square of the Mahalanobis distance, and T stands for transposition. For multidimensional inputs, one can compute the variance for each dimension separately as well as calculate the correlation between the dimensions (covariance). The resulting covariance matrix represents a description of the shape and orientation of the data, taking the first and second order moments of the data into account. See the EllipseCalculations.xls in the Supporting Information as a demonstration of how this representation can be calculated from an ellipse specified by its center point, two half-axes and rotation.

Gating-ML also supports decision tree structures where a binary decision tree is stored for the gate. The decision tree encodes a sequence of computing steps that are supposed to be applied on an event in order to decide about its membership in the decision tree gate. A dimension and a threshold are specified in each non-terminal node of the tree. In each computing step, the value of the event is compared against the threshold. Based on the result, computing continues in the "less than" or "greater or equal" tree branch, respectively. The membership results are encoded in terminal nodes. This form of a gate is computationally fast to process; it allows for a complete specification of any arbitrary multidimensional region, whether contiguous or not, and it also allows for encoding of a gate when the boundary is difficult to define geometrically.

Finally, Boolean collections of any of the types of gates extend the expressiveness of the Gating-ML language. Any number of arbitrary gates may be combined using the "AND", "OR", and "NOT" operators to describe complex multidimensional regions and to combine gates defined in different dimensions. The operand gates can either be defined in line or a gate reference may be used.

Built-in Data Transformations

Gating-ML includes built in public transformations that have been shown useful for display or analysis of cytometry data. These include *logarithmic*, *polynomial of degree one* (i.e. linear combination with translation), *square root*, *asinh* (inverse hyperbolic sin), *split-scale* (22), *Hyperlog* (21), and *ratio of two parameters*, as well as inverse transformations wherever these exist, i.e. *exponential*, *quadratic transformation*, *hyperbolic sin*, *inverse split scale*, and *EH* transformations (21). The inverse transformations are useful when transformed data are only available; including cases where list mode data are stored on logarithmical scales in FCS files. The exponential transformation may be used to describe the conversion of channel values to linear scale prior to applying any further transformation, such as (20–22).

Hyperlog (21) is a transformation that performs linear-like for low and negative values and log-like for high values. It is defined as inverse of a linear combination of an exponential with a linear transformation, and is more suitable for certain kinds of FCM data analysis compared to a traditional logarithmic scale. Depending on implementation, the inverse-based nature of the Hyperlog definition may make it computationally expensive. If performance is an issue, the asinh and the split-scale transformations may be used as simple and computationally inexpensive alternatives.

Not all Gating-ML data transformations are reversible, which supports the design choice to describe gates in transformed space. For example, linear combination of parameters is a transformation, which actually loses some of the original information; however, it is used in analytical software to "increase" the number of parameters that can be visually inspected in a two-dimensional space. Another non-reversible transformation is the ratio of two parameters, which becomes useful for computational depolarizations and normalizations, such as using the forward scatter to normalize fluorescence values.

Compensation

In FCM, the emission spectral overlap of fluorescent labels makes it usually necessary to correct detected signals before using the values as a basis for other analyses. Fluorescence compensation is the process by which the amount of spectral overlap is subtracted from the total detected signals to yield an estimate of the actual amount of each dye. Within Gating-ML, fluorescence compensation is a type of transformation. Although compensation of a single parameter can also be described as a linear combination, storing spillover matrices is more transparent and it defines compensation of all involved parameters at once (i.e. as a function from n to n arguments). Gating-ML supports spillover matrices and referencing compensated parameters individually.

Initial Software Implementations

Support for Gating-ML has been included in the *flowUtils* R package. The R Project for Statistical Computing (25) is an open-source research platform for evaluating and implementing statistical methods. To support statistical analysis in FCM several R packages (*flowCore* (32), *plateCore*, *flowUtils*, *flowQ*, *flowClust* (26), and *flowViz*) are being developed within Bioconductor (27,28), an open source and open development R-based software project for the analysis and comprehension of genomic data.

We have also developed an open source Java application named FACEJava in order to test the implementability of the Gating-ML specification. This operating system independent tool is capable of applying gating, compensation and other transformations on FCS list mode data files. FACEJava aided the development process as it identified implementation bottle necks and other issues that were subsequently solved during the early stage of development of Gating-ML. It has been invaluable in both, testing design aspects of this specification and building compliance tests as informative references for third-party developers.

Discussion

FCS is an essential standard assuring FCM data interoperability. Although there is a minor overlap in information that can be captured in FCS and Gating-ML, they are complementary standards. FCS will continue to be used to primarily encode the raw measurements, and Gating-ML is intended only to capture post-acquisition analysis. Both FCS and Gating-ML will be maintained by ISAC and further adapted to future requirements.

The basis of the Gating-ML design is a compromise providing enough expressive power to describe gating and traditional two-dimensional analyses that are common in existing software, while attempting to accommodate future innovations in automated multidimensional gating (and clustering) in a generic way. Wherever appropriate, this has been implemented in a consistent manner on both, syntactic and semantic level. Uniform referencing of parameters and transformations from all types of gates can serve as an illustrative example. However, usefulness has been chosen over consistency if "unavoidable". For example, all but the rectangle gates are boundary including whereas rectangle gates are minimum-including and maximum-excluding. This aspect is a direct reflection of received feedback from the community requesting to be able to unambiguously split negatively vs. positively expressed markers although covering the whole range of values. We feel we have achieved a balance in the expressiveness and simplicity of Gating-ML that will enable it to easily be adopted by the developer community.

Gates described according to Gating-ML are exchangeable among compliant software applications. The mathematical accuracy of the Gating-ML specification ensures that any compliant software will provide the same results when applying gates. The specification has been designed with respect to computational simplicity of gate processing. Specifically, the description of gates was designed to facilitate simple computational decisions regarding whether a particular event is within a particular gate. The receiver of gate descriptions does not need any complicated algorithms to apply gates, which simplifies development of compliant software. For example, the description of convex polytope gates as intersection of half-spaces is computationally less expensive and algorithmically much simpler compared with the alternative approach of computing convex hulls of finite sets of points. Also, the description of ellipsoid gates using covariance matrices is significantly easier and faster to process compared to approaches modeling ellipsoid axes in multidimensional space.

Initial drafts included the possibility of describing any kind of transformation in an embedded MathML (29) segment. This would allow for a greater level of flexibility; however, it has been shown during development that the adoption costs of such a specification would be too high. Although there are many software tools for displaying MathML, there are significant issues with computing MathML expressions. Therefore, MathML is only used to provide additional formal documentation for each transformation type in the *transformations XML schema*, not for expressing transformations in Gating-ML files.

Gating-ML CR does not include any transformations or other components that we believe may require the use of subject matter covered by patents or other intellectual property rights, or that may only be available under restrictive licensing conditions. The DSTF followed the *Requirements for a data file standard format to describe cytometry and related analytical cytology data* (30), including the specific requirements stating that *Implementation of the standard shall be possible without charge* and *Implementation of the standard shall be non-restrictive.* A consequence of this approach is the lack of support for the Logicle/Bi-exponential transformation (20) within Gating-ML as it is covered by a patent (31) and is licensed under restrictive conditions. With the continued evolution of the FCM field, it is to be expected that new analysis techniques will be developed and that Gating-ML CR will require future extensions to reflect these advancements. Gating-ML was therefore designed to be easily extendible. If necessary, new gate and transformation types can be added consistently with existing gates and transformations. It is suggested that any future extension employ the name of the XML schema that is to be extended as the prefix of that of the extension with the period character, ".", as the separator.

Conclusion

Gating-ML strives to be an open standard made available to the general public, as developed, approved, and maintained via a collaborative and consensus driven process under the auspices of ISAC. It is an essential standard that will enable the exchange of analysis between software tools, accelerating flow cytometry discovery in the same way that FCS was such a success in describing the raw data. With the open specification and the provision of informative compliance tests, we aim to provide a path for low and zero-cost implementations to be validated. Although the significant features are mostly locked, community input is requested on the suitability of the specification and how implementable the standard is from the developer's perspective. With the W3C-like stepwise approval process, ISAC is trying to ensure that Gating-ML serves its purposes and meets well the business needs of all involved parties. Although the specification may slightly change further in the future, significant features and design aspects will likely remain unchanged. The purpose of this CR is to elicit feedback from the scientific community on its suitability, and from the instrument vendor and third-party software tool community as to how implementable the standard is. Based on feedback to the DSTF the standard will be modified as appropriate before being submitted to ISAC Council as a candidate for ISAC Recommendation.

Acknowledgments

RRB is a Research Scholar of Michael Smith Foundation for Health, and an ISAC Scholar. James Wood is especially thanked for chairing the ISAC DSTF review process.

Literature Cited

1. Gasparetto M, Gentry T, Sebti S, O'Bryan E, Mimmanapalli E, Blaskovich MA, Bhalla K, Rizzieri D, Haaland P, Dunne J, Smith C. Identification of compounds that enhance the anti-lymphoma activity of rituximab using flow cytometric high-content screening. J Immunol Methods 2004;292:59–71.
2. Chicurel M. Bioinformatics: Bringing it all together. Nature 2002;419:751, 753, 755.
3. Lizard G. Flow cytometry analyses and bioinformatics: Interest in new softwares to optimize novel technologies and to favor the emergence of innovative concepts in cell research. Cytometry 2007;71A:646–647.
4. Boddy L, Wilkins MF, Morris CW. Pattern recognition in flow cytometry. Cytometry 2001;44:195–209.
5. Herzenberg LA, Parks D, Sahaf B, Perez O, Roederer M, Herzenberg LA. The history and future of the fluorescence activated cell sorter and flow cytometry: A view from stanford. Clin Chem 2002;48:1819–1827.
6. Data file standard for flow cytometry. Data file standards committee of the society for analytical cytology.Cytometry 1990;11:323–332.
7. Seamer LC, Bagwell CB, Barden L, Redelman D, Salzman GC, Wood JCS, Murphy RF. Proposed new data file standard for flow cytometry, version FCS 3.0. Cytometry 1997;28:118–122.
8. ISAC. ISAC E-NEWS—March 2005. Available at: http://www.isac-net.org/index.php?option=com_content&task=view&id=364&Itemid=119.
9. Maecker HT, Rinfret A, D'Souza P, Darden J, Roig E, Landry C, Hayes P, Birungi J, Anzala O, Garcia M, Harari A, Frank I, Baydo R, Baker M, Holbrook J, Ottinger J, Lamoreaux L, Epling CL, Sinclair E, Suni MA, Punt K, Calarota S, El-Bahi S, Alter G, Maila H, Kuta E, Cox J, Gray C, Altfeld M, Nougarede N, Boyer J, Tussey L, Tobery T, Bredt B, Roederer M, Koup R, Maino VC, Weinhold K, Pantaleo G, Gilmour J, Horton H, Sekaly RP. Standardization of cytokine flow cytometry assays. BMC Immunol 2005;6:13.
10. Spidlen J, Gentleman RC, Haaland PD, Langille M, Le Meur N, Ochs MF, Schmitt C, Smith CA, Treister AS, Brinkman RR. Data standards for flow cytometry. OMICS 2006;10:209–214.
11. Lee J, Spidlen J, Boyce K, Cai J, Crosbie N, Dalphin M, Furlong J, Gasparetto M, Goldberg M, Goralczyk EM, Hyun B, Jansen K, Kollmann T, Kong M, Leif RC, McWeeney S, Moloshok TD, Moore W, Nolan G, Nolan J, Nikolich-Zugich J, Parrish D, Purcell B, Qian Y, Selvaraj B, Smith C, Tchuvatkina O, Wertheimer A, Wilkinson P, Wilson C, Wood J, Zigon R; the International Society for Advancement of Cytometry Data Standards Task Force, Scheuermann RH, Brinkman RR. MIFlowCyt: Minimum Information about a Flow Cytometry Experiment. Cytometry Part A 2008; 73A: in press. DOI: 10.1002/cyto.a.20623.
12. World Wide Web Consortium (W3C). Extensible Markup Language (XML). Available at: http://www.w3.org/TR/REC-xml/.
13. W3C. XML Schema. Available at: http://www.w3.org/XML/Schema.
14. Leif RC. CytometryML, binary data standards. Manipulation and Analysis of Biomolecules, Cells, and Tissues II. SPIE Proc 2005;5699:325–333.
15. Shadbolt N, Berners-Lee T, Hall W. The semantic web revisited. IEEE Intell Syst 2006;21:96–101.
16. Spidlen J. International Society for Analytical Cytology Data Standards Task Force, Brinkman R. Analytical Cytometry Standard – Gating-ML Component (text of the specification, January 20, 2008). Available at: http://www.isac-net.org/media/standards/Gating-ML.v1.5.080120.pdf. Accessed November 03, 2008.
17. Spidlen J, International Society for Analytical Cytology Data Standards Task Force, Brinkman R. Analytical Cytometry Standard – Gating-ML Component (full specification, January 20, 2008). Available at: http://www.isac-net.org/media/standards/Gating-ML.v1.5.080120.full.zip. Accessed November 03, 2008.
18. Object Management Group (OMG). Unified Modeling Language (UML), version 2.0. Available at: http://www.omg.org/technology/documents/formal/uml.htm.
19. Spidlen J. Gating-ML Compliance Tests. Available at: https://sourceforge.net/project/showfiles.php?group_id=175725&package_id=202309. Accessed October 6, 2008.
20. Parks DR, Roederer M, Moore WA. A new "logicle" display method avoids deceptive effects of logarithmic scaling for low signals and compensated data. Cytometry 2006;69A:541–551.
21. Bagwell CB. Hyperlog-a flexible log-like transform for negative, zero, and positive valued data. Cytometry 2005A;64:34–42.
22. Battye FL. A Mathematically Simple Alternative to the Logarithmic Transform for Flow Cytometric Fluorescence Data Displays. Available at: http://www.wehi.edu.au/cytometry/Abstracts/AFCG05B.html.
23. Sunday D. Geometry Algorithms—Algorithm 3—Point in a Polygon. Available at: http://www.geometryalgorithms.com/Archive/algorithm_0103/algorithm_0103.htm. Accessed March 06, 2008.
24. Wikipedia. Mahalanobis distance. Available at: http://en.wikipedia.org/wiki/Mahalanobis_distance. Accessed June 18, 2008.
25. The R Foundation, The R Development Core Team. The R Project for Statistical Computing. Available at: http://www.r-project.org/. Accessed July 31, 2007.
26. Lo K, Brinkman RR, Gottardo R. Automated gating of flow cytometry data via robust model-based clustering. Cytometry 2008;73A:321–332.
27. Gentleman R, Carey V, Huber W, Irizarry R, Dudoit S, editors. Bioinformatics and Computational Biology Solutions using R and Bioconductor. New York: Springer; 2005.
28. Bioconductor—Open Source Software for Bioinformatics. Available at: http://www.bioconductor.org/. Accessed July 31, 2007.
29. W3C Math Working Group. Mathematical Markup Language (MathML) Version 2.0, 2nd ed. W3C Recommendation. Available at: http://www.w3.org/TR/MathML2/. Accessed July 31, 2007.
30. Spidlen J, Leif RC, Brinkman RR. Requirements for a data file standard format to describe cytometry and related analytical cytology data. Available at: http://downloads.sourceforge.net/flowcyt/Requirements-v070920.pdf. Accessed March 08, 2008.
31. Parks DR, Moore W. Methods and systems for data analysis, USPTO Application# 20060015291. Available at: http://www.freshpatents.com/Methods-and-systems-for-data-analysis-dt20060119ptan20060015291.php. Accessed March 08, 2008.
32. Hahne F, LeMeur N, Brinkman RR, Ellis B, Haaland P, Sarkar D, Spidlen J, Strain E, Gentleman R. flowCore: A Bioconductor software package for high throughput flow cytometry data analysis. Cytometry A; in press.