applied biosystems
by life technologies

Extensible Sequence (XSQ) File Format Specification 1.0.1

Table of Contents

1 Introduction

This document describes a new extensible file format for storing sequence data. It will initially be used as the format for the 5500 output file, which can have multiple independent readings at the same position in a fragment for sequencing. This file contains primary analysis results for a single lane in a flowcell. The standard file extension for the file is ".xsq", but the brief name XSQ will often be used to refer to this output file format.

The format is a specific schema based on the more general HDF5 format, which is a hierarchical binary format that supports the concepts of groups, datasets, and attributes. These concepts are very similar to the concepts of directories (or folders), files, and properties of a typical hierarchical computer file system. See the HDF Group's website for more information about the HDF5 format. From that website one can also obtain the free Java-based tool *HDFView* to browse any HDF format files, including XSQ.

2 File Format

The schema is described in detail in the following numbered outline. The following typefaces as used for different objects within the file:

- **Groups are expressed in bold typeface**
- *Attributes are written in italic typeface*
- Datasets with compound datatypes are in normal roman typeface
- Datasets with uniform datatypes are underlined

If no attributes are mentioned for a group, it will possess no attributes other than standard HDF attributes. For names of groups, datasets, and fields that may be user defined, the relevant variable name is enclosed in angle brackets, e.g. <**TagName**>, <OtherCallQV>, or <LibraryName>.

1 **RunMetadata**
The RunMetadata group stores data about the run as individual attributes.

1.1 *FileVersion* [String (255); Mandatory (hardcoded)]
This attribute defines the version of the XSQ file format specification.

1.2 *HDFVersion* [String (255); Mandatory (hardcoded)]
This attribute defines the version of the HDF framework used.

1.3 *AnalysisSoftware* [String (255); Mandatory; Default: "Unknown"]
This attribute describes the version of the software used on the instrument for data collection and primary analysis during the creation of this XSQ file.

1.4 *InstrumentVendor* [String (255); Mandatory; Default: "Unknown"]
This attribute describes the name of the instrument manufacturer.

1.5 *InstrumentModel* [String (255); Mandatory; Default: "Unknown"]
This attribute describes a specific platform (e.g., "5500/5500XL") used for this run.

1.6 *InstrumentSerial* [String (255); Mandatory; Default: "Unknown"]
This attribute contains a unique serial number to identify a particular instrument.

1.7 *InstrumentName* [String (255); Mandatory; Default: <InstrumentSerial>]
This attribute describes the lab-defined name of a particular instrument (Harry, Ron, Hermione).

1.8 *LaneNumber* [8-bit unsigned integer; Mandatory; Default: 255]
This attribute describes the lane number within the flowchip layout in which this sequencing sample was loaded.

1.9 *FlowcellAssignment* [8-bit unsigned integer; Mandatory; Default: 255]
This attribute describes the position of the instrument flowcell in which this flowchip was placed.

1.10 *RunName* [String (255); Mandatory; Default: <InstrumentName>_<RunStartTime>]
This attribute contains the name of a flowchip run.

1.11 *SequencingSampleName* [String (255); Mandatory; Default: "Unknown"]
This attribute contains the name of the sample loaded in a lane on a flowchip.

1.12 *SequencingSampleDescription* [String (255); Optional]
This attribute contains the description of the sample loaded in a lane on a flowchip.

1.13 *LibraryType* [String (255); Mandatory; Default: "Unknown"]
This attribute contains the description of library preparation protocol used to prepare the sample for sequencing. { 'MatePair' | 'PairedEnd' | 'Fragment' }

1.14 *RunStartTime* [String (255); Mandatory]
This attribute contains the run start time in the format yyyy-mm-dd hh:mm:ss.

1.15 *RunEndTime* [String (255); Mandatory; Default: RunStartTime]
This attribute contains the run end time in the format yyyy-mm-dd hh:mm:ss.

1.16 *SequencingCenter* [String (255); optional]
This attribute contains the name of the sequencing center where the instrument is located.

1.17 *Operator* [String (255); Mandatory; Default: <UserName>]
This attribute contains the name or ID of the person setting up the instrument run.

1.18 *FileUUID* [String (255); Optional; Generated]
This attribute contains a universally unique identifier for this file.

1.19 *IsIndexingRun* [8-bit unsigned integer; Mandatory]
This attribute describes whether indexing was used to classify multiple libraries within the run. (It is interpreted as a Boolean.) A value of zero means that no indexing was used; any non-zero value implies indexing.

1.20 *ConversionToolAndVersion* [String(255); Optional]
This attribute contains the name and version of the tool that was used to generate this XSQ file from FASTA/CSFASTA files.

1.21 *ConversionInputFiles* [String(255); Optional]
This attribute contains a semicolon separated list of files with paths that were used to generate this XSQ file from FASTA, FASTQ, CSFASTA, QUAL, and/or CSFASTQ files.

1.22 *ConversionCommand* [String(255); Optional]
This attribute contains the full command, including executable, files, options, and their values used to generate this XSQ file from FASTA/CSFASTA files.

1.23 FragmentFilteringStrategy [Optional]
This table contains up to eight text descriptions to describe the fragment filtering dataset -- one description for each of the 8 bits used in the filtering dataset. It is only available if filtering is applied in the Fragments group. It contains 3 fields:

- BitPosition [8-bit unsigned integer] describes which bit is used to show pass or fail status for this filter (0=pass, 1=fail).
- FilterName [String (255)] is a name referring to this filter.
- Description [String (255)] is a description of the filter's properties.

1.24 TagFilteringStrategy [Optional]
This table contains up to eight text descriptions to describe the tag filtering datasets -- one description for each of the 8 bits used in the filtering dataset. It is only available if filtering is applied in any Tag group. It contains 3 fields:

- BitPosition [8-bit unsigned integer] describes which bit is used to show pass or fail status for this filter (0=pass, 1=fail).
- FilterName [String (255)] is a name referring to this filter.
- Description [String (255)] is a description of the filter's properties.

1.25 LibraryDetails [Mandatory]
This table describes the properties of each library used in this file. It contains 10 fields:

- LibraryName [String (255)] describes the name of one library preparation, which may be used across several indexes in an indexing run.
- Application [String (255)] describes how this sequence is intended to be analyzed. { 'Whole Genome Resequencing' | 'Targeted Resequencing' | 'Whole Transcriptome (Fragment)' | 'Whole Transcriptome (PairedEnd)' | 'Small RNA' | 'ChIP-Seq' | 'Methylation' }
- ProjectName [String (255)] describes the project within which this sample is being run.
- SampleOwner [String (255)] is the owner of this sample.
- SampleIdentifier [String (255)] is a unique name for this sample.
- Description [String (255)] describes the sample.
- Comments [String (255)] stores additional information about this sample or its preparation.
- ERCC [8-bit unsigned integer] determines whether internal controls have been used (0=no, 1=yes).
- Species [String (255)] is the intended species for mapping. { 'Homo sapiens' | 'Mus musculus' | 'other' | …}
- Assembly [String (255)] is the intended assembly for mapping (may be blank if unknown). { 'hg18' | 'hg19' | 'mm9' | …}
- LibraryInsertSizeMinimum [32-bit unsigned integer]
  This attribute contains the expected minimum insert size in nucleotides. It is used when paired end or mate pair libraries are used, and set to zero for fragment libraries. (May be zero if unknown.)
- LibraryInsertSizeMaximum [32-bit unsigned integer]
  This attribute contains the expected maximum insert size in nucleotides. It is used when paired end or mate pair libraries are used, and set to zero for fragment libraries. (May be zero if unknown.)

1.26 TagDetails

This group contains a child node for each tag (read type) that describes its tag-level properties.

1.26.1 <TagName>

This group contains a set of attributes describing the basespace and colorspace data for a single type of read.

1.26.1.1 *TagSequence* [String (255); Conditional: required when colorspace is present]
This is the full sequence of the tag used for primer annealing. It is used in conjunction with the read offset to determine the nucleotide or nucleotides for phasing the colorspace reads. When the full sequence is not available, the last base(s) of the tag are sufficient as long as they cover the largest offset in the DatasetColorEncoding table.

1.26.1.2 *IsColorPresent* [8-bit unsigned integer; Optional; Default 0]
This attribute describes the presence or absence of colorspace call data. (It is interpreted as a Boolean.) When no calls are reported in colorspace (and thus the ColorCallQV dataset is missing), the value of this attribute is zero. When color calls are reported, the value is non-zero.

1.26.1.3 *IsBasePresent* [8-bit unsigned integer; Optional; Default 1]
This attribute describes the presence or absence of base call data. (It is interpreted as a Boolean.) When no calls are reported in base space (and thus the BaseCallQV dataset is missing), the value of this attribute is zero. When base calls are reported, the value is non-zero.

1.26.1.4 *NumBaseCalls* [32-bit unsigned integer; Conditional]
This attribute describes the maximum untrimmed read length in base space across all reads for this tag. This is the width of the BaseCallQV dataset. It is not used when the BaseCallQV dataset is not present, but is required when BaseCallQV is present.

1.26.1.5 *MinTrimmedReadLength* [32-bit unsigned integer; Mandatory]
This attribute describes the minimum trimmed read length across all reads for this tag (NumCalls – TrimEndLength – TrimStartLength).

1.26.1.6 *DoesRepresentedStrandMatchSource* [8-bit unsigned integer; Optional; Default 1]
This attribute describes whether the reads are represented on the same strand as the source. (It is interpreted as a Boolean.) Zero if the synthesis strand is not the same as the source strand, 1 otherwise.

1.26.1.7 *RelativeOrder* [8-bit unsigned integer; Conditional]
This attribute describes the relative order of a read on the sense strand of the source nucleic acid in relation to all of the other reads from the same fragment. 1 based. Required only when there is more than one tag per fragment.

1.26.1.8 *SynthesisFromThreePrime* [8-bit unsigned integer; Optional; Default 0]
This attribute describes the synthesis direction of the synthesized strand. (It is interpreted as a Boolean.) A zero or missing value indicates that the synthesis direction is from the 5' end toward the 3' end. Any non-zero value indicates synthesis from the 3' end to the 5' end.

Values for current library types are described here:

Instrument	LibraryType	Indexing	TagName	Base	DoesRepresentedStrand MatchSource	RelativeOrder	SynthesisFrom ThreePrime
SOLiD™	Fragment	Yes | No	F3	T	1	1	1
SOLiD™ + 5500	Mate Pair	No	F3	T	1	2	1
			R3	G	1	1	1
SOLiD™	Paired End	No	F3	T	1	1	1
			F5-P2	T	0	2	0
SOLiD™	Paired End	Yes	F3	T	1	1	1
			F5-BC	G	0	2	0
5500	Paired End (DNA)	Yes | No	F3	T	1	1	1
			F5-DNA	T	0	2	0
5500	Paired End (RNA)	Yes | No	F3	T	1	1	1
			F5-RNA	G	0	2	0
Illumina	Paired End		Forward	N/A	0	1	0
			Reverse	N/A	1	2	0

1.26.1.9 DatasetColorEncoding [Conditional: required when colorspace is present]
This table describes how reads are encoded as colors by the instrument. Indexing reads are not included in this table. This table has the following columns.

- DataSetName [String (255)] must match the name of a read dataset under a tag group.
- Offset [8-bit signed integer], as detailed in figure 2, describes the distance and direction of the primer end as compared to the tag end, which is used to relate individual calls to the bases they encode.
- Encoding [String (255)] contains a set of integers that direct the conversion from nucleotides to colors as described below. (2BE is '11', 4BE is '1303').
- Stride [8-bit unsigned integer] is used for packing discontinuous reads into a continuous array; it is used to determine the relative positioning of the start nucleotides between color calls. (The use of stride disallows collating calls from multiple offsets into a single call vector when an incomplete set is used, so these are separated into different datasets.)
- NumColorCalls [32-bit unsigned integer] describes the maximum untrimmed read length in base space across all reads for this tag; this is the width of the associated CallQV dataset.

Default values for 2BE and 4BE are described in the "Generalizations and Extended Specification" section below.

1.27 *ExternalFragmentIdUsed* [8-bit unsigned integer; Optional; Default 0]
This attribute describes whether the fragments have an external FragmentId that should be used to represent the fragment, as is the case when yxLocation is not available. It is interpreted as a Boolean, where zero represents absence of the FragmentId dataset and 1 declares that the FragmentID dataset is present and should be used.

2 Indexing [Conditional]
This group only exists for multiplexing runs. It contains index tag reads and QVs as well as classification (indexing) information for each fragment. There is one subgroup for each image unit of a given lane.

2.1 LibraryIndex [Mandatory]
This table stores the index ID, the index name, and its association with a library. It has these columns:

- IndexID [16-bit unsigned integer] is a nonnegative integer that is unique within the file. Zero is reserved for the Unclassified library and all other libraries have unique positive IndexID values (but are not necessarily consecutive).
- IndexName [String (255)] is the name of the index from the indexing kit (Index1). IndexName values are unique within the LibraryIndex table and have a 1:1 mapping with IndexID.
- LibraryName [String (255); alphanumeric plus underscore and space; first character alpha] is the user-defined name of the library (e.g. Larry).

It is possible that multiple IndexIDs are associated with one LibraryName (e.g. Larry_Index1, Larry_Index2). IndexID values are used to store index classification in the IndexClassification dataset.

2.2 *IndexKitName* [String (255); Optional]
The attribute contains the name of the kit used to add the indexing nucleotides.

2.3 *IndexLength* [32-bit unsigned integer; Mandatory]
The attribute contains the length of the index reads. Abbreviated L_I.

2.4 **<ImageUnitID>**
This set of groups partitions the fragment reads by imaging unit (panel or tile), and is included primarily for systems that partition the imaging area into multiple units (e.g., 'Panels'). The name of each unit is called an Imaging Unit and has an associated ID. The name of each group instance is identical to the four-digit string used to identify the image unit with which the contained data is associated (e.g. "0058"). Each of the group's three datasets are sorted in the same order and have multiplicity equal to the number of total fragments.

2.4.1 *FragmentCount* [32-bit unsigned integer; Mandatory]
This attribute contains the total number of fragments (e.g., 'Beads') reported in the image unit. Abbreviated N_I.

2.4.2 yxLocation [2 x 16-bit unsigned integer; Mandatory]
This N_I-by-2 dataset contains fragment positions, which are in "y,x" coordinates. Fragment positions are sorted by coordinate y, then by coordinate x.

2.4.3 ColorCallQV [8-bit unsigned integer; Mandatory]
This N_I-by-L_I dataset gives the color call and QV for indexing reads of all fragments. Both the color call and the quality value are written as described in the "Color Call and QV" section below.

2.4.4 IndexClassification [16-bit unsigned integer; Mandatory]
This N_I-by-1 dataset gives the index classification information for each fragment of a given image unit.

3 <LibraryName>_<IndexName>, 'Unclassified', or 'DefaultLibrary'

This group contains results – base space and/or color space calls – for a set of fragments. There are two distinct use cases: runs using indexing vs. runs that do not use indexing.

For files derived from runs *without indexing*, the **DefaultLibrary** group exists but the others do not, as there is no need to partition fragments in a non-indexed run. The name of the library will be written as an attribute to this group (described below), but the name of this group is fixed as 'DefaultLibrary'.

For files derived from runs *with indexing*, each index generates its own group; a separate **Unclassified** group exists for the fragments that could not be accurately assigned to any index group. (The **DefaultLibrary** group does not exist.) The unclassified fragments are retained in case the wrong indexing scheme was used and the index classifications must be re-generated. It is possible that multiple Indices are associated with one specimen (e.g. Larry_Index1, Larry_Index2, Moe_Index3, Moe_Index4, Curly_Index5, Curly_Index6), but indices will be unique across all libraries. In an indexing run, each **<LibraryName>_<IndexName>** group contains either all or a subset of the image units that are found in the **Indexing** group; when an image unit is missing (such as when the focus fails), the data for that set of reads cannot be generated and that image unit may be missing.

3.1 *LibraryName* [String (255); Mandatory; alphanumeric plus underscore and space; first character alpha]
This attribute matches the LibraryName in the LibraryName_IndexName group name and also matches a library in the LibraryIndex table of the Indexing group for indexing libraries.

3.2 *IndexName* [String (255); Conditional: only used with indexed libraries]
This attribute contains the index name from the indexing library kit. It is the same as the IndexName in the LibraryName_IndexName group name. It may be different from the IndexID.

3.3 *IndexID* [32-bit unsigned integer; Conditional: only used with indexed libraries]
This attribute contains a unique integer with a 1:1 mapping to an IndexName. Zero is reserved for the Unclassified library; all other groups have unique positive IndexID values (but may not be consecutive).

3.4 *LibraryIndexUUID* [String (255); Optional; generated]
This attribute contains a universally unique identifier for the LibraryIndex node to confirm association with external LIMS.

3.5 <ImageUnitID>

For systems that partition the imaging area into multiple units (e.g., 'Panels'), the name of each unit is called an Imaging Unit, and each has an associated ID represented as a four-digit string (e.g., "0058"). Systems that do not use imaging or use a single image unit have a single group here named "0000". When no fragments are found in an image unit, that image unit will not be created. Each such group contains a group named "Fragments", and one or more tag groups as described below.

3.5.1 *FragmentCount* [32-bit unsigned integer; Mandatory]
The attribute describes the number of fragments in the image unit. Abbreviated as N_f.

3.5.2 Fragments

This group contains fragments that belong to an image unit <ImageUnitID>. Each such group contains an yxLocation dataset and an optional Filtering dataset. The fragment ordering is preserved between the datasets of the Fragments group and the TagName groups (below).

3.5.2.1 yxLocation [2 x 16-bit unsigned integer; Conditional]
This N_f-by-2 dataset contains fragment positions, which are in "y-x" coordinates. Fragment positions are sorted by coordinate y, then by coordinate x. At least one of the yxLocation or FragmentID datasets must be present. When yxLocation is used, it must be present for all fragments.

3.5.2.2 Filtering [8-bit unsigned integer; Optional]
This dataset of N_f elements gives the filtering information at the fragment level. Any non-zero value corresponds to a particular filtering strategy.

3.5.2.3 *NumFragmentsPassed* [32-bit unsigned integer; Optional]
An attribute that indicates how many fragments pass all filters (having value of zero in the Filtering dataset) across all tags.

3.5.2.4 FragmentId [String(255); Conditional]
This dataset of N_f elements stores fragment names when they can not be derived from ImageUnit and yxLocation. At least one of the yxLocation or FragmentID datasets must be present. When FragmentId is used, it must be present for all fragments.

3.5.3 <TagName>

The image unit group can contain one or more of following tag groups, named "F3", "F5", "R3", "R5", [and other labels for other technologies, such as 'forward' and 'reverse', or pulse group order for strobe sequencing, or read order for multiple reads of the same template]. Each group contains the read data of a particular tag for fragments in this image unit.

3.5.3.1 *NumReadsPassed* [32-bit unsigned integer; Optional]
This attribute indicates how many reads pass all filters (having value of zero in the Filtering dataset).

3.5.3.2 BaseCallQV [8-bit unsigned integer; Conditional]
This N_f-by-NumBaseCalls dataset gives the color call and QV for reads from this TagName. Both the color call and the quality value are written as described in the "Color Call and QV" section below. The 'BaseCallQV' dataset name is reserved for base call information only, and is not required to appear in the DatasetColorEncoding table. This dataset is not required when all calls are in colorspace.

3.5.3.3 ColorCallQV [8-bit unsigned integer; Conditional]
This N_f-by-NumColorCalls dataset gives the color call and QV for reads from the main set of color calls. While the 'ColorCallQV' name is reserved to describe this data from the first complete set of primers collated together, its name is still listed in the DatasetColorEncoding table (in the TagDetails group). Both the color call and the quality

value are written as described in the "Color Call and QV" section below. This dataset is not required when colorspace data is not generated.

3.5.3.4 <OtherCallQV> [8-bit unsigned integer; Conditional]
This N_f-by-NumColorCalls dataset gives the color call and QV for reads from additional runs beyond the initial set in the ColorCallQV dataset. Multiple datasets are allowed, but each must be described with a unique name in the DatasetColorEncoding table in the TagDetails group. This dataset is not required when additional sets of colorspace data are not generated.

3.5.3.5 Filtering [8-bit unsigned integer; Optional]
This dataset of N_f elements gives the filtering information in a tag level, where any bit set at 1 represents flagging by a single filter. Any non-zero value of the 8 bits taken together corresponds to a filtering by one or more particular filtering strategies.

3.5.3.6 TrimStartLength [16-bit unsigned integer; Optional]
This dataset of N_f elements describes the number of bases to be trimmed from the start of the read.

3.5.3.7 TrimEndLength [16-bit unsigned integer; Optional]
This dataset of N_f elements describes the number of bases to be trimmed from the end of the read.

3 Generalizations and Extended Specification

Generalizing colorspace usage requires describing the details of positioning the calls as well as their encoding from nucleotide combinations into colorspace.

Primer, Probe, and Call Positioning

The abstraction of the priming, probe annealing, and ligation process can be described as follows:

1. Each tag has a unique name and a known sequence
2. Each tag has a set of primers that anneal with varying offsets from the primer's end
3. For some primers, bridge (unlabeled, dark) probes are used to reach beyond the end of the tag
4. The positioning of the primer end and the presence of a bridge probe determine the offset for the resulting calls.

Determining the positioning of the start of the call vector in relation to the tag is necessary to combine calls from various primer and probe cycles on the same fragment. (See figure 1.)

The variables necessary for to describe this positioning are

1. Tag Name
2. Tag Sequence
3. Set of Primers
    a. Primer Name
    b. First Probe Offset (adjusted for presence of bridge probe)

For each primer, a cycle of probe annealing, detection, and cleavage are performed. For standard (2BE), calls from the first complete set of primers are collated into a single call vector for simplified processing. For subsequent primer runs, the calls are coalesced into a more compact vector by removing the uncalled locations; these are expanded back to their original positions by using the probe length (described below). Probe encoding details are described next.

Colorspace Encodings

Historical colorspace encodings have used only two base encoding (2BE) with a simple encoding scheme of G=(1,1) as described in the whitepaper. This section describes changes to include a generalization of the encoding process to support future encodings and specific details for the new four base encoding (4BE, where 4 is the length of the vector, not the number of informative sites) with G=(1,3,0,3). This can be described with the following formula (given the matrix multiplication and matrix addition as described below) as illustrated in the example:

$$\text{Color} = \Sigma_i \, [g_i * \text{int}(\text{nucleotide}_i)]$$

As described in the whitepaper and in figure 3 below, the encoding of nucleotide sequences into colorspace follows several steps:

1. Mapping nucleotides to integers.
    a. A=>0, C=>1, G=>2, T=>3

2. Multiplication of integer-encoded nucleotides by their encoding value using a defined multiplication matrix. Multiply "CTGG" => "1322" by G=(1,3,0,3) to get "1201"
    i. C=>1; 1*1 => 1
    ii. T=>3; 3*3 => 2
    iii. G=>2; 2*0 => 0
    iv. G=>2; 2*3 => 1
3. Use ordered iterations of addition matrix on "1201" to get "2"
    i. 1+2=3
    ii. 3+0=3
    iii. 3+1=2

The variable definitions needed for this encoding are

1. Encoding Name (2BE, 4BE for defaults or other name for non-default)
2. Encoding Scheme (G=[1,3,0,3])
3. Nucleotide to Integer table (A=>0, C=>1, G=>2, T=>3)
4. Multiplication Matrix (see figure 3)
5. Addition Matrix (see figure 3)
6. Length of probe remaining after color cleavage

Call and QV Packing

Calls and Quality Values are encoded and compressed in a single byte per call, whether that call is a base call or in colorspace. These values are written as a byte array in the order they are read from the instrument. Note that this is 5' to 3' in most cases, but is 3' to 5' for SOLiD reads with 5' probes (F5 and R5).

Each call and its associated quality value are stored in a compressed form in a single byte. The call, either bases transformed into integers (A=>0 or 00; C=>1 or 01; G=>2 or 10; T=>3 or 11) or calls in four color space represented by integers 0-3 are written in the 2 least significant bits. The quality value is written in the six higher bits. For the QV, the range of the 6 bits is between 0 and 63. Valid quality values will be from 3 to 62, represented as PHRED scores [QV = $-10*\log_{10}P(error)$]. 0 and 1 are reserved for future usage; 2 indicates a missing quality value (not trimming); 63 refers to an uninformative or missing call.

Calls and Quality values can be unpacked from a single byte using this example code in syntax for C:

```
Call = CallQV & 0x03;
QV   = CallQV >> 2;   // 0, 1, 2, and 63 are special
```

Default Values

	2BE	4BE
TagName	F3, F5, R3, R5	F3, F5, R3, R5
ProbeEncoding	1,1	1,3,0,3
Probe Length	5	5
FirstProbeOffset	-1 (Collated)	1
NucleotideCode	A 0 C 1 G 2 T 3	
MultiplicationMatrix		
AdditionMatrix		

MultiplicationMatrix

	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	3	1
3	0	3	1	2

AdditionMatrix

	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

4 Figures

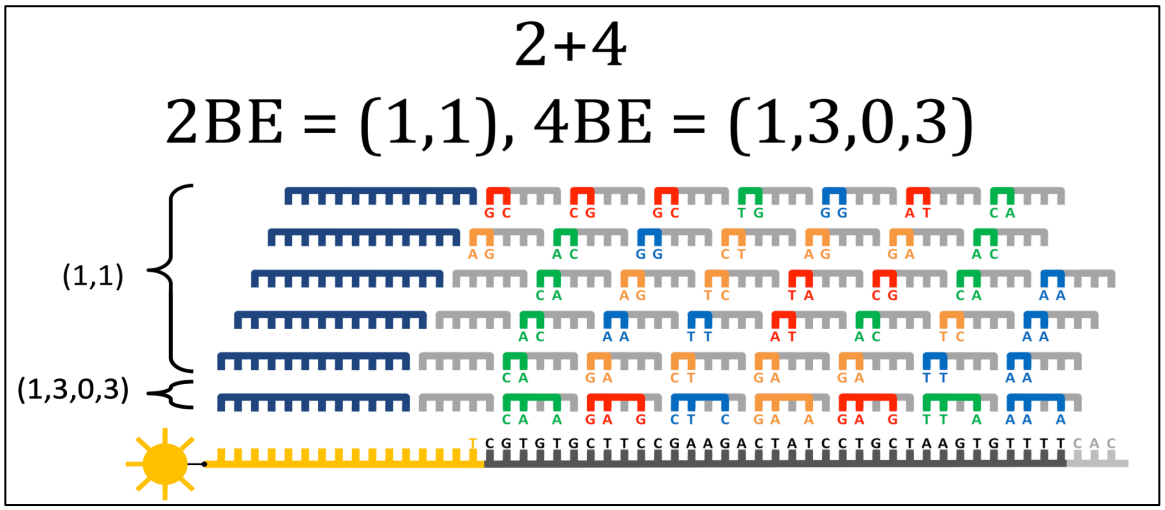

Figure 1. Graphical description of standard 2 base encoding [2BE; G=(1,1)] with additional error correction primer [4BE; G=(1,3,0,3)] for error correction.

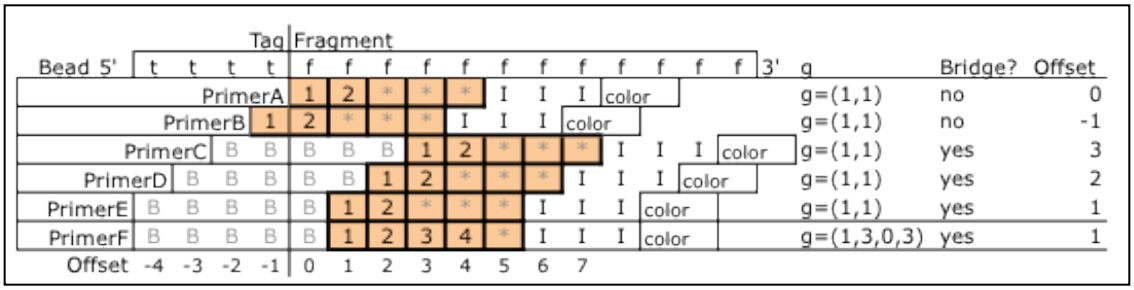

Figure 2. Detailed description of primer and probe locations. Top row shows the Bead with an attached DNA fragment, where 't' depicts the sequencing tag and 'f' shows the beginning of the DNA fragment to be sequenced. Primers A-E use probes that have 2 base encoding (12***) and primer F uses 3 base encoding (1234*). Note use of bridge (dark) probes with primers C-F. Offsets show the difference between the end of the tag and the beginning of the first non-dark probe ligation.

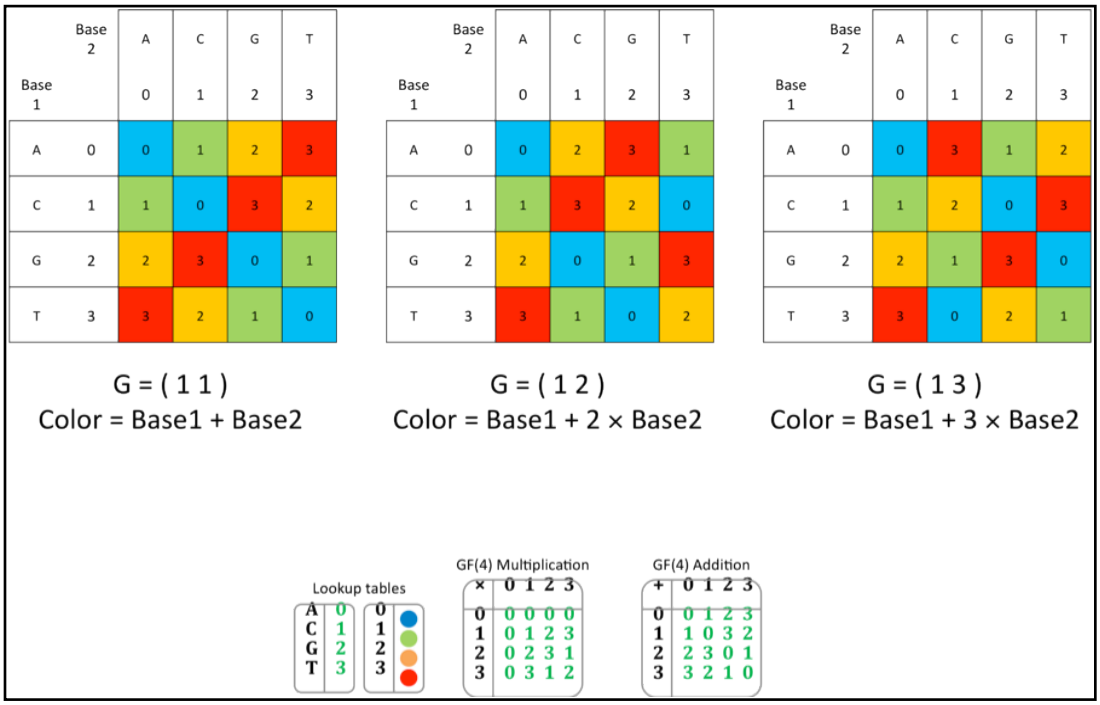

Figure 3. Examples of 2 base encoding with three different encoding vectors [G=(1,1), G=(1,2), G=(1,3)]. Note the tables at the bottom of the figure for integer and color lookup.

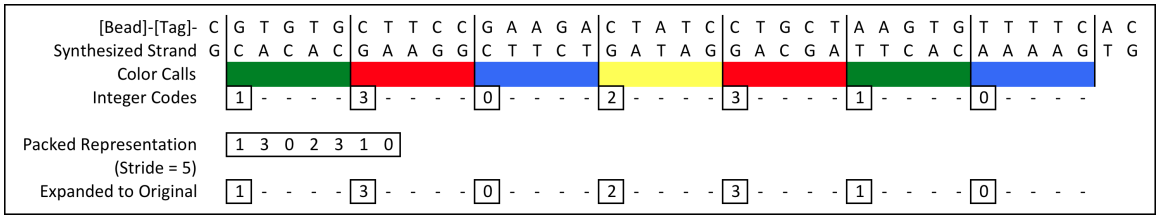

Figure 4. Schematic view of the use of STRIDE to pack discontinuous calls. This is the same sequence from the first figure, showing a single primer run of the last primer (4BE). The color calls are read at 5 base intervals, and their integer codes are shown at the beginning of every probe. As there is consistent missing information for 4 of the 5 bases, the data can be represented more compactly by representing only the calls at known positions; the distance between positions is called the *stride*. The packed representation can be converted back to the original by using the stride to space the existing calls appropriately.

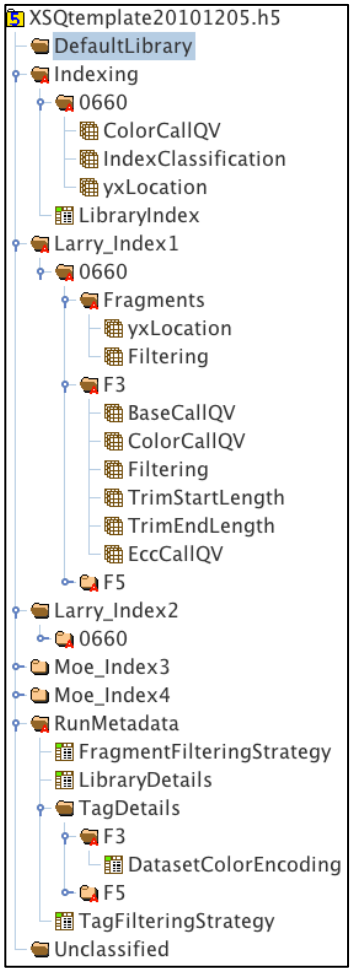

Figure 6. HDFView screenshot of the XSQ file format, showing Groups and Datasets.

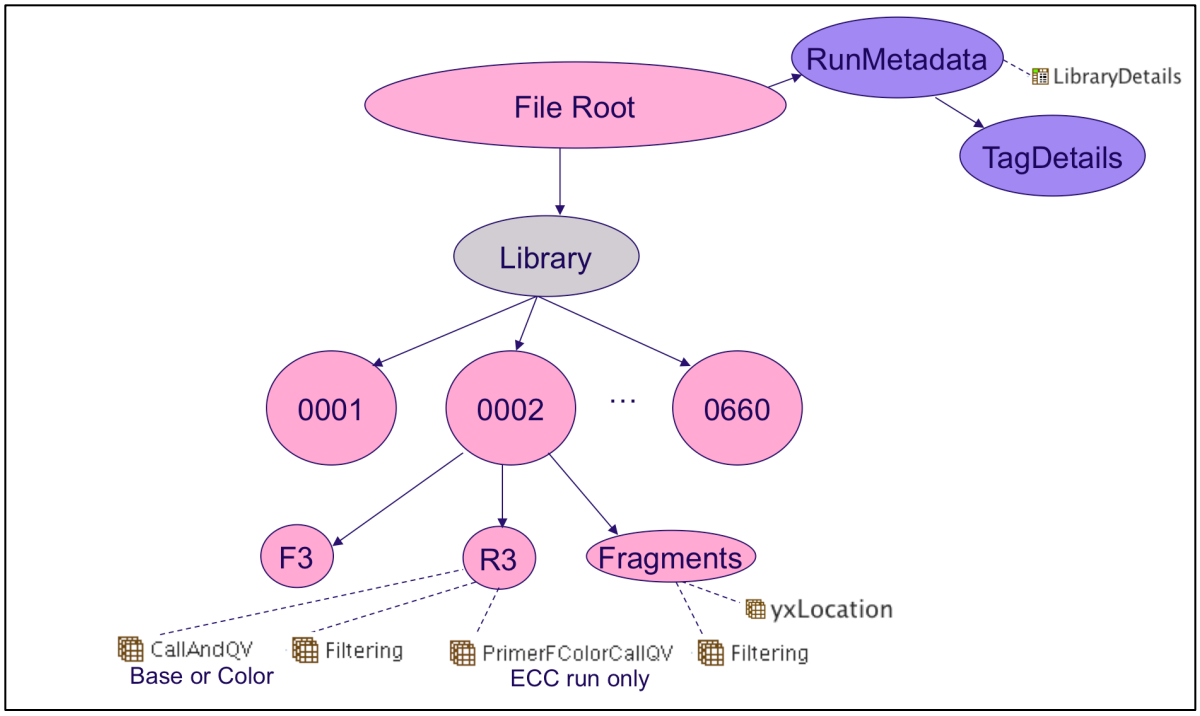

Figure 5. Schematic view of the XSQ file format without Indexing.

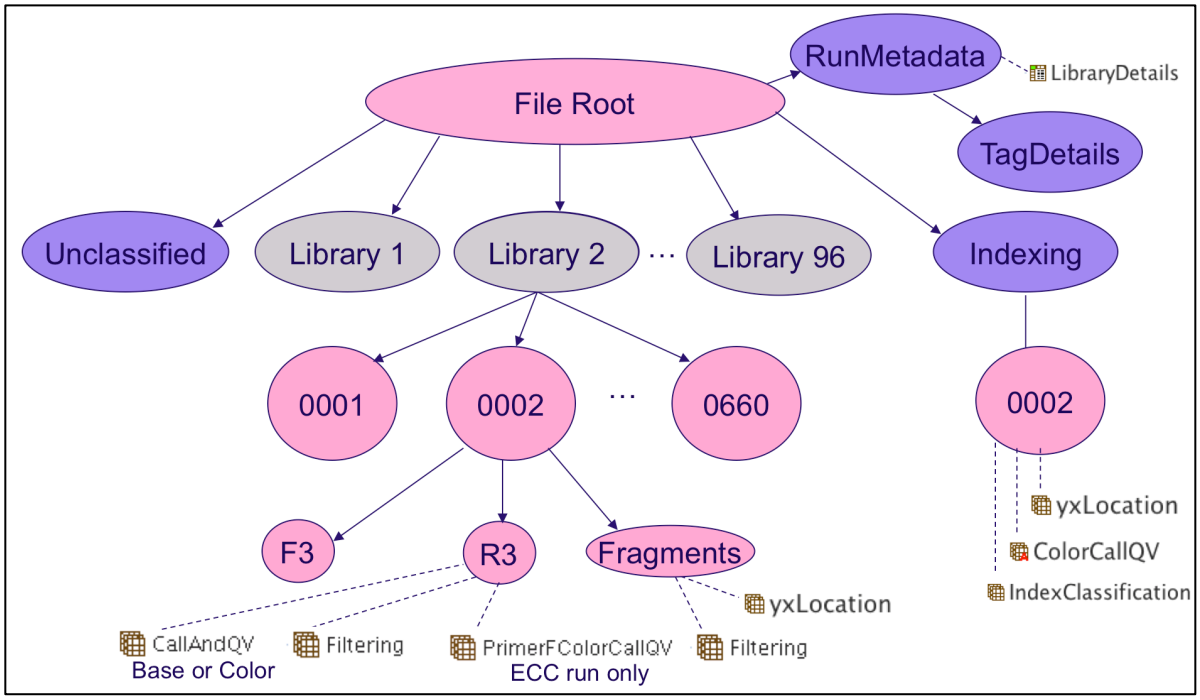

Figure 6. Schematic view of the XSQ file format with Indexing.

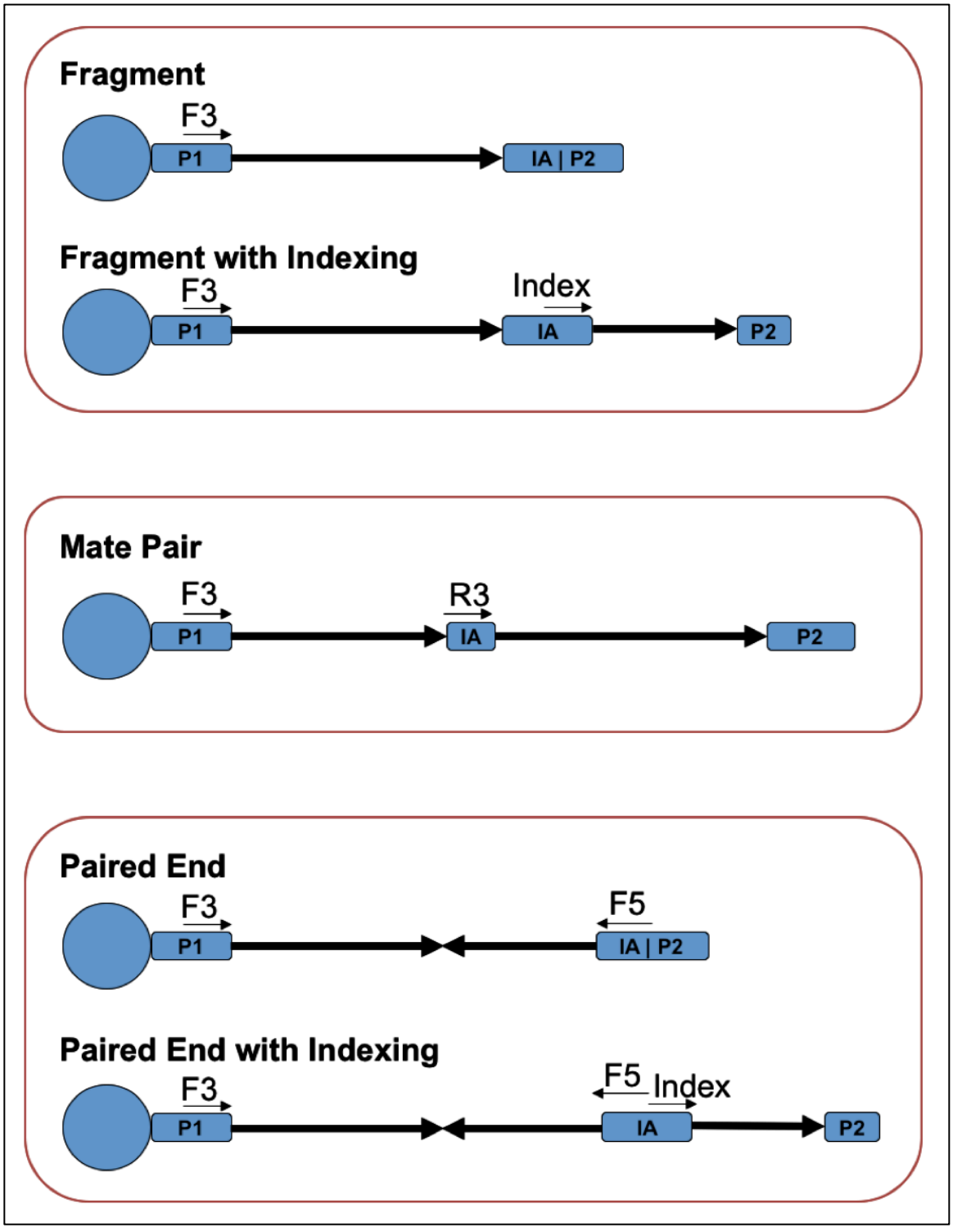

Figure 7. Life Technologies SOLiD™ Library constructs.

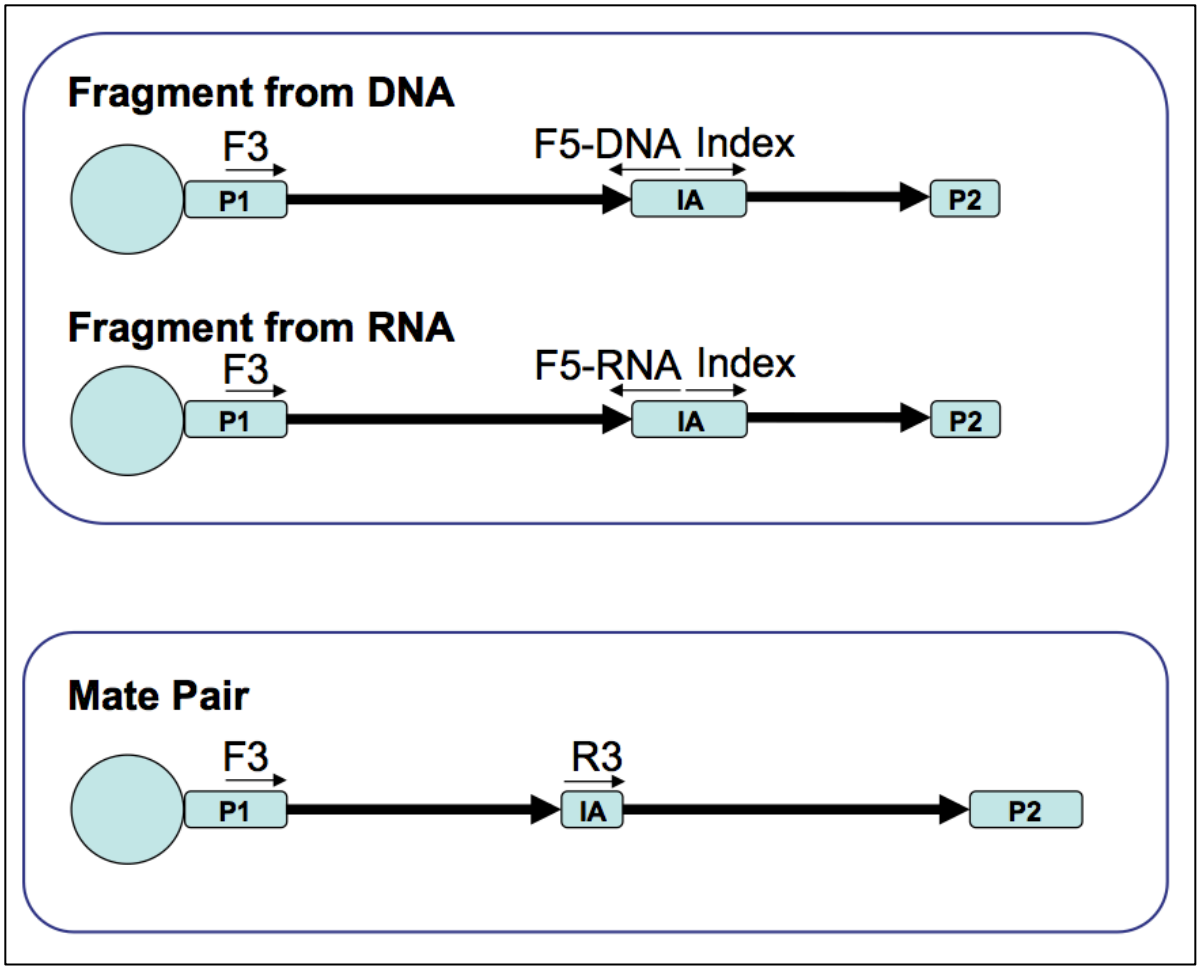

Figure 8. Life Technologies 5500 Library constructs.