Scaffold Batch Annotated SCAFML File

(Batch XML Driver File)

Revision 2020-09-15

Proteome Software, Inc

Scaffold Batch version 4.11+

Available at:

http://www.proteomesoftware.com/scaffold-batch-annotated-scafml

Annotated XML Driver (SCAFML) File Documentation

The following XML driver file includes references to the batch processing version of Scaffold (Scaffold Batch) for most elements and attributes. To learn more about the functionality associated with the XML instructions, please refer to the available Scaffold and Scaffold Q+/Q+S documentation (also included with your Scaffold installation by going to **Help > Help Contents…**). Note that, where possible, the default values are used in this example to be efficient with space. For example, if the element contains an attribute with a value, that value is probably the default that would be used if not set at all. In some cases, different values will be used and are noted for clarity. ***Note: This document is not intended to be copied into a SCAFML and used to process data with Scaffold Batch.***

Syntax Conventions

Please note that for consistency, Unix-style forward slashes for path designations, and bash-style command syntax will be used throughout this document.

Command line options

To view command line interface options built in to the Scaffold Batch command, run it without arguments:

```
user@host:~$ ./ScaffoldBatch4

   ScaffoldBatch [-q] [-f] [-keypath PATH] [xmlDriverFile1] [xmlDriverFile2]
      Options:
         -h, -help, --help
            Shows help documentation
         -q, -quiet, --quiet
            Write minimal information to console
         -f, -force, --force
            Doesn't request answers from user
         -k, -keypath, --keypath PATH
            Specifies location of the license key path -- useful for grid systems.
         --is-license-valid-at PATH
            Checks if license is valid at the given path
```

You can also get some help information by running with the help option (see Appendix A for more details):

```
user@host:~$ ./ScaffoldBatch4 --help
```

Annotated XML Driver (SCAFML)

XML Code	Associated Functionality
`<Scaffold>`	The pair of tags **`<Scaffold>`** to **`</Scaffold>`** brackets all the other specifications in the XML file. It is common to add the version of Scaffold here, eg: `<Scaffold version="Scaffold_4.8.0">`
`<Experiment name="Demo"`	The pair of elements **`<Experiment>`** to **`</Experiment>`** brackets the rest of the specification.
`description="An Example Experiment"`	Menu action: **Experiment > Edit Experiment**

containsCommonContaminants="false"	
load="existing.sf3"	Menu action: **File > Open**

peakListGeneratorName="Bioworks"
peakListGeneratorVersion="3.2"
peakListDeisotoped="false"
peakListChargeStatesCalculated="false"
peakListGeneratorAnnotation="false"
	Publications View: entries specified by user

showStatisticsPane="false"
showDataLoadingPane="false"
showSamplesPane="false"
showProteinsPane="false"
showProteinHomologyPane="false"
showQuantifyPane="false"
showPublishPane="false"
showStatisticsPane="false"
	Menu action: **Edit > Preferences**
	These options allow the View buttons, along the left in Scaffold GUI to be displayed. **"false"** would hide this View. Default for these is **"true"**.

condenseDataWhileLoading="true"	**Scaffold Wizard > New BioSample > Condense data as it is loaded** checkbox

displayType="Exclusive Spectrum Count"
clusters="collapsed"
	Valid values for **displayType**:
	1. **Protein Identification Probability**
	2. **Percent Coverage**
	3. **Percentage of Total Spectra**
	4. **Exclusive Unique Peptide Count**
	5. **Total Unique Peptide Count**
	6. **Exclusive Unique Spectrum Count**
	7. **Total Unique Spectrum Count**
	8. **Exclusive Spectrum Count**
	9. **Total Spectrum Count**
	10. **Quantitative Value**
	Valid values for **clusters**: **expanded** or **collapsed**

password="foo"
protectThresholds="true"
protectDisplaySettings="true"
	Menu action: **Edit > Preferences**

`unimodFile="/path/to/file"`	Use this to specify alternate Unimod file. **NOTE:** **`unimodFile=""`** (empty string) is equivalent to Scaffold GUI option **Edit > Preferences > Paths > Do not use UNIMOD.**

`analyzeWithTandem="true"`
`analyzeWithSubsetDB="true"`
	Scaffold Wizard > Load and Analyze Data > Analyze with X! Tandem options

`highMassAccuracyScoring="false"`
`useIndependentSampleGrouping="false"`
`use3xScoring="false"`
`useFamilyProteinGrouping="true"`
`proteinGrouping="protein-cluster-analysis"`
	Turn on High Mass Accuracy Scoring
	Protein Grouping
	Turn on Independent Sample Grouping
	Use old 3.x scoring
	Note: for new v4 LFDR scoring, set **`highMassAccuracyScoring`** and **`use3xScoring`** both to **`"false"`**.
	Clustering Options
	Valid values:
	1. `protein-cluster-analysis`
	2. `experiment-wide`
	3. `independent-sample`
	NOTE: See Appendix B for more information. We still maintain backward compatibility with legacy attributes. For example, using **`proteinGrouping=experiment-wide`** is the same as if you had the attributes **`useFamilyProteinGrouping=false`** and **`useIndependentSampleGrouping=false`**

`connectToNCBI="true"`
`annotateWithGOA="true"`
	Menu action: **Experiment > Add NCBI Annotations** (Note: This option is set by default to **`"false"`**)
	Menu action: **Experiment > Add GO Annotations**, which annotates with GOA database

`Merge="/path/to/file.sf3">`	Menu action: **File > Merge with SFD/SF3**

```
<FastaDatabase
    id="db0"
    name="generic"

    databaseAccessionRegEx=">([^ ]*)"
    databaseDescriptionRegEx=">[^ ]* (.*)"

    decoyProteinRegEx="REVERSE|RANDOM|-R|##"
    path="/path/to/sprot_human.fasta"

    databaseVersion="1.0" />
```

If the XML driver contains parse rules, they will be used both in reading the data and in applying the protein sequences;

If the XML driver contains no parse rules but the database has been previously indexed (either in Batch or GUI) the existing index will be used in both phases;

And, finally, if no parse rules are included in the XML driver and the database has not been indexed already, Scaffold will make its best guess at a correct parse rule or use a generic rule.

The id attribute of the **`<FastaDatabase>`** tag will be matched to the **`database`** attribute of the **`<BiologicalSample>`** element.

We write out **`databaseVersion`** to the **`FastaDatabase`** element when creating a SCAFML. However, functionality for reading that attribute had been in place since January 2013.

```
<FastaDatabase
    databaseAccessionRegEx=">([^\s]*)"
    databaseDescriptionRegEx=">[^\s]*[\s](.*)" id="db0"
    path="/path/to/db0.fasta" />

<FastaDatabase
    databaseAccessionRegEx=">([^\s]*)"
    databaseDescriptionRegEx=">[^\s]*[\s](.*)" id="db1"
    path="="/path/to/fileb.fasta " />

<MultiFastaDatabase id="m1">
    <FastaRef id="db0" />
    <FastaRef id="db1" />
</MultiFastaDatabase>
```

To support multiple FASTA databases, use the **`MultiFastaDatabase`** element with **`FastaRefs`** after you have declared individual databases. The **`FastaRef`**'s id should reference a database declared prior to it. The order that databases are used is from top to bottom, in this example **`db0`** will be searched first followed by **`db1`**. **`BioSample`** elements can now reference this multi-database using the **`id`** of **`MultiFastaDatabase`**.

Note the two **`FastaDatabase`** elements, corresponding to the new element, **`MultiFastaDatabase`** element.

```
<ExperimentSetup>
    <MaxQuant
        autoGenerateBiosamplesFrom="experiment"
        database="db0"
        quantitativeTechnique="Stable Isotope Labeling"
        condenseData="true"
        analyzeAsMudpit="false"
        directory="/path/to/dir">
    </MaxQuant>

    <MascotDistiller
        projectDirectory="/path/to/dir/with-rov-xml">
    </MascotDistiller>

    <ProteomeDiscoverer
        msf="/path/to/msf"
        load="individual-biosamples"
        quantitativeTechnique="Spectrum Counting"
        analyzeAsMudpit="false"
        thinDataAsLoaded="true"
    </ProteomeDiscoverer>
```

The **`<ExperimentSetup>`** element can handle MaxQuant, Mascot Distiller, and Proteome Discoverer data, but is flexible enough of accommodate other data types in the future.

`autoGenerateBiosamplesFrom="experiment"`

Batch will load each MaxQuant experiment as a separate BioSample

`autoGenerateBiosamplesFrom="fraction"`

This will create a separate BioSample for each fraction.

For Discoverer, the `load` attribute can have one of two values:

`"individual-biosamples"`

or

`"one-biosample"`

`</ExperimentSetup>`
 	
`<Modification unimodName="HNE" />`
	Note that the attributes: **`database`**, **`quantitativeTechnique`**, **`analyzeAsMudpit`** and their values will be applied to each of the auto generated BioSamples
	
	Load Biosample Wizard
	
	Select a predefined modification.
	Specify extra modifications beyond those in the search results data file. These extra modifications will be added onto the modifications used by the search engine.
	Additionally, specify modifications when doing X! Tandem analysis. If not, then no extra modifications need be specified aside from extra modifications mentioned above.
	
	The list of `unimodName`s for standard modifications in the Unimod list can be seen in the Load BioSample wizard.

`<Modification fullName="testMod"`
`    referenceName="a test mod"`
`    monoMass="42"`
`    averageMass="42"`
`    aminoAcidsModified="A" />`
	Create a new modification not already in Unimod. For use with X! Tandem search when the modification is not one of the default modifications already in the Unimod list.
	
	Note: From the "Analyze Data with X! Tandem" screen, choose the New button.
	For N- and C-terminal modifications, use lower case **`"n"`** and **`"c"`**.

`<BiologicalSample`
`    analyzeAsMudpit="false"`
`    database="db0"`
`    name="Demo_01"`
`    category="Category 1">`
`    <InputFile>/path/to/results</InputFile>`
`    <InputFile>Demo_01.dat</InputFile>`
`    <InputFile>Demo_01.tar.gz</InputFile>`
`</BiologicalSample>`
	The pair of tags **`<BiologicalSample>`** to **`</BiologicalSample>`** enclose the input files that are to be loaded into this biological sample.
	The database is required. It is defined with the **`<FastaDatabase>`** tag.
	The **`database="db0"`** attribute attaches the database to the file being loaded.
	Multiple input file types can be loaded.
	Scaffold Batch 4 recognizes files from various search engines. Check this compatibility matrix for more details.

```
<PurityCorrections>
    <PurityCorrection
        id="iTRAQ 4-PLEX Purity Correction"
        quantitativeTechnique="iTRAQ 4-Plex">
         0.000,0.000,0.000,0.000,0.000
         0.000,0.000,0.000,0.000,0.000
         0.000,0.000,0.000,0.000,0.000
         0.000,0.000,0.000,0.000,0.000
    </PurityCorrection>
</PurityCorrections>

<MultiplexExperimentalDesign
    analysisType="ratio"
    experimentType="common-reference"
    quantitativeTechnique="iTRAQ 4-Plex">

    <ReferenceAlignmentGroup id="0">
        <Category id="Category 1">
            <TechnicalReplicateGroup>
                <QuantitativeSample
                    msID="1"
                    name="Quant 2"
                    reporter="iTRAQ-114"
                    description="" />
            </TechnicalReplicateGroup>
            <QuantitativeSample
                msID="2"
                name="Quant 4"
                reporter="iTRAQ-116"
                description=""/>
        </Category>
        <Category id="Reference">
            <TechnicalReplicateGroup>
                <QuantitativeSample
                    msID="3"
                    name="Quant 1"
                    reporter="iTRAQ-115"
                    description=""/>
            </TechnicalReplicateGroup>
        </Category>
    </ReferenceAlignmentGroup>
</MultiplexExperimentalDesign>

<BiologicalSample
    category="iTRAQ"
    description=""
    name="iTRAQ Sample 1"
    quantitativeTechnique="iTRAQ 4-Plex"
    purityCorrection="iTRAQ 4-PLEX Purity Correction">
<UnorganizedSamples>
     <QuantitativeSample msID="4"
         name="Quant 3"
         reporter="iTRAQ-117"
         description=""/>
</UnorganizedSamples>
    <InputFile>/path/to/input/data</InputFile>
</BiologicalSample>


 <QuantificationSetup
    blockingLevel="Unique Peptides"
    calculationType="Mean" minimumDynamicRange="0.01"
    normalizationBetweenSamples="true"
    referenceType="Individual Spectrum Reference"
    spectrumQualityFilter="No filter"
    useNonExclusivePeptides="false">
</QuantificationSetup>
```

Improved Q+/Q+S support in Scaffold Batch

Valid values for **MultiplexExperimentalDesign**’s type attribute are:

1. **independent-groups** (Between-subjects (Independent Groups))
2. **common-reference** (Between-subjects (Common/Pooled Reference))
3. **repeated-measures** (Repeated Measures / Time Course)

Some general notes:

Every child element for a **ReferenceAlignmentGroup** is a category

QuantitativeSamples are childless

You must nest **QuantitativeSamples** in a **TechnicalReplicateGroup** tag to indicate a technical replicate group

QuantitativeSamples without a **TechnicalReplicateGroup** parent tag are simply singleton samples

The id attribute is only required to be unique in the element name space in which they are given, that is, a **ReferenceAlignmentGroup** and **QuantitativeSample** may have the same id, but no two **ReferenceAlignmentGroups** may have the same id.

QuantitativeSample can be a sibling or descendant of **TechnicalReplicateGroup**

A **TechnicalReplicateGroup** cannot be a child of **QuantitativeSample**

A reference category, must have the literal `id` **“Reference”**

BiologicalSample element will have a **quantitativeTechnique** attribute. The following are valid values, anything else is classified internally as **UNKNOWN**:

```
Spectrum Counting
iTRAQ 4-Plex
iTRAQ 8-Plex
TMT 2-Plex
TMT 6-Plex
TMT 10-Plex
Stable Isotope Labeling Multiplex
TIC
Precursor Intensity
```

Each **PurityCorrection** element should have a unique id and values, it will be an error otherwise.

Because of the fact that we can load multiple quantitative techniques per BioSample and Q+ can only run one type, **MultiplexExperimentalDesign** has an attribute specifying which to use.

Support for Quantitative settings

Menu action (In Q+/Q+S): **Quant > Quantitative Settings**

```
<MultiplexQuantitativeSettings
    foldChangeReferenceCategory=""
    fwerCorrectionType="No Correction" id="default"
    multiplexTest="Kruskal-Wallis Test"
    showOnlyAnalyzedSamplesInView="true"
    significanceLevel="0.1">
    <QuantitativeSample
        description="" msID="1" name="Quant 3"
        reporter="iTRAQ-116"/>
    <QuantitativeSample
        description="" msID="1" name="Quant 4"
        reporter="iTRAQ-117"/>
    <QuantitativeSample
        description="" msID="1" name="Quant 1"
        reporter="iTRAQ-114"/>
</MultiplexQuantitativeSettings>
```

Menu action (In Q+/Q+S): **Quant > Quantitative Testings**

```
<Export thresholds="thresh"
    type="statistical-analysis-report"
    multiplexQuantSettingsId="default"/>
```

To export with appropriate settings, you must use the `id` from `MultiplexQuantitativeSettings`, eg:

`multiplexQuantSettingsId="default"`

```
<DisplayThresholds name="Some Thresholds"
    id="thresh"
    proteinProbability="0.95"
    minimumPeptideCount="2"
    peptideProbability="0.95"
    minimumPeptideLength="0"
```

Menu action: **Edit > Edit Peptide Thresholds**

The attributes `proteinProbability`, `minimumPeptideCount` and `peptideProbability` correspond to the filter thresholds at the top of the Samples and Proteins pages. These thresholds can be applied to the exported data. See the Export section below.

By itself **`<DisplayThresholds.../>`** filters the proteins displayed.

The id attribute defined here is applied to the **`"thresholds=..."`** attribute of the Export element below. This can be utilized if multiple exports require different export filter settings. Thus, multiple **`DisplayThresholds`** entries can be used here.

If the **`<DisplayThresholds>`** term is missing or parameters undefined, these default thresholds are used:

`proteinProbability="0.99"`

`peptideProbabilty="0.95"`

`minimumPeptideCount="2"`

`minimumPeptideLength` is a "custom" threshold that doesn’t show peptides shorter than "X."

```
    useDeltaMassTolerance="true"
    deltaMassTolerance="100"
    useAMU="false"/>
```

`useDeltaMassTolerance`: "custom" threshold to enable setting delata mass tolerances at the parent ion level.

`deltaMassTolerance` is also a "custom" threshold to set the actual mass tolerance used in filtering.

`useAMU` is a "custom" threshold to specify the units of the **`deltaMassTolerance`**: true for AMU or false for PPM.

<DisplayThresholds
name="Some Thresholds"
id="thresh"
minimumNTT="1"
useCharge="true,true,true,true"
proteinProbability="0.5"
minimumPeptideCount="2"
peptideProbability="0.5">
	The pair of tags **<DisplayThresholds>** to **</DisplayThresholds>** bracket the optional custom thresholds for specific search engines.
	Some General Minimum Thresholds (**Edit > Edit Peptide Thresholds...**) include **minimumNTT** and **useCharge**, representing minimum tryptic termini and whether to use charge states +1, +2, +3 or +4 and higher.
	The child elements below are equivalent to selecting the individual program thresholds in the Configure Peptide Thresholds dialog box.

<MascotThresholds
ionMinusIdentityScore="0"
ionScores="20.0, 20.0, 30.0, 40.0"/>
	The other terms are for custom thresholds that can be set from the Configure Peptide Thresholds panel.

<SequestThresholds
xCorrs="1.8, 2.5, 3.0, 3.5"
deltaCn="0.1"/>
	Note that scores for each charge state are listed separately.

<TandemThresholds logExpectScores="2.0, 2.0, 2.0, 2.0"/>	Note that scores for each charge state are listed separately.
<ZCoreThresholds zcoreScores="100,100,100,100"/>	Note that scores for each charge state are listed separately.

<PhenyxThresholds
peptideProbability="90,90,90,90"
zScore="5,5,5,5"/>
	Note that scores for each charge state are listed separately.

<OmssaTresholds
peptideProbability="95,95,95,95"
logExpectScore="2,2,2,2"/>
	Note that scores for each charge state are listed separately.

<ProteinLynx
peptideProbability="95,95,95,95"
score="30,30,30,30"/>
	Note that scores for each charge state are listed separately.

```
<DisplayThresholds
    name="Thresholds"
    id="thresh"
    maxProteinFDR="0.01"
    maxPeptideFDR="0.01"

</DisplayThresholds>
```

Specify FDR from Scaffold Batch

For `DisplayThresholds` element, users will use the attributes **`maxProteinFDR`** and **`maxPeptideFDR`** where the value must be in the range [0, 1].

Note that when in the Scaffold GUI, and you set FDR values for protein/peptide thresholds, if you export a SCAFML, it will set the `DisplayThresholds` element with the corresponding **`peptideProbability`** and **`proteinProbability`** values, not the **`maxProteinFDR`** and **`maxPeptideFDR`**.

```
<Export
    type="sf3"
    thresholds="thresh"
    saveOnlyIdentifiedSpectra="true"
    saveNoSpectra="true"
    discardBelowThresholds="false"
    discardSpectraWithPeptidesCount="2"
    saveFrozen="true"
    path="."/>
```

Menu action: **File > Save Condensed Data…**

This saves the data loaded into Scaffold Batch as an SF3 file. This file can be read by Scaffold or by the free Scaffold Viewer.

Note: When saving sf3, **`discardSpectraWithPeptidesCount`** and **`discardUnlessModifiedWith`** are applied only when **`discardBelowThresholds="true"`** and we have a reference to **`<DisplayThresholds>`** element.

The attribute **`thresholds="..."`** refers back to the threshold **`id`** defined in **`<DisplayThresholds>`**.

`saveOnlyIdentifiedSpectra="true"` saves only the identified spectra in your Scaffold SF3 file.

`saveNoSpectra="true"` saves the SF3 file without spectra.

`discardBelowThresholds="false"` will discard unneeded spectra. Equivalent to "save as condensed."

`discardSpectraWithPeptideCount`: discards spectra on proteins with more than "X" peptides, to save file space. MCP requires that all one-hit-wonders have spectra available, but other proteins don't need spectra to back up the identification. This is equivalent to setting of "1."

The **`saveFrozen`** setting saves the file as frozen, where users can't change thresholds.

```
<Export
    type="sf3"
    thresholds="thresh"
    saveCondensed="identified-spectra"
    path="."/>
```

For backwards compatibility, here are the combinations to get the same functionality as the ***newly*** added **`saveCondense`** option:

`saveCondensed="identified-spectra"`
equivalent to:

```
  saveNoSpectra="false"
  discardSpectraWithPeptidesCount="-1"
  saveFrozen="false"
```

`saveCondensed="frozen-identified-spectra"`
equivalent to:

```
  saveNoSpectra="false" thresholds="threshold-id"
  discardSpectraWithPeptidesCount="-1"
  saveFrozen="true"
```

`saveCondensed="no-spectra"`
equivalent to:

```
  saveNoSpectra="true"
```

	`discardSpectraWithPeptidesCount="0"`
	`saveFrozen="false"`
	
	`saveCondensed="frozen-no-spectra"`
	equivalent to:
	`saveNoSpectra="true"`
	`thresholds="threshold-id"`
	`discardSpectraWithPeptidesCount="0"`
	`saveFrozen="true"`
	
	`saveCondensed=mcp-required`
	equivalent to:
	`saveNoSpectra="false"`
	`discardSpectraWithPeptidesCount="2"`
	`discardUnlessModifiedWith="Carbamidomethyl,`
	`Oxidation"`
	`saveFrozen="false"`
	
	`saveCondensed="frozen-mcp-required"`
	equivalent to:
	`saveNoSpectra="false" thresholds="threshold-id"`
	`discardSpectraWithPeptidesCount="2"`
	`discardUnlessModifiedWith="Carbamidomethyl,`
	`Oxidation" saveFrozen="true"`

`<Export`
`type="protxml"`
`thresholds="thresh"`
`path="file.protxml"/>`
	This saves the data loaded into Scaffold Batch as a **protXML** file. This is a data format for loading into a database.

`<Export`
`type="mzIdentML"`
`version="1.1.0"`
`showDecoys="false"`
`useFilter="true"`
`individualReports="false"`
`useGzip="false"`
`writePeaklists="true"`
`path="/path/to/file.mzid"`
`threshold="thresh"/>`
	This saves an **mzIdentML** file (with extension **MZID**) for use in other software. This format is becoming more widely used in the proteomics community.
	
	Note, version option that takes 1.0 or 1.1, eg, will accept anything that starts with 1.0 or 1.1 like 1.1.0
	
	Note that to export with filters (and thus the current filter, as described in the PTM mzIdentML export setting); see the `<DisplayThresholds>` section above.
	
	The **mzIdentML** export setting "Include peaklists" is true by default. Generally, all examples are written for the default value.
	
	Note that MZIDs can be exported without spectra if nothing passes filter, but are not valid **mzIdentML** files.

```
        <Export
            type="experiment-report"
            thresholds="thresh"
            useNormalization="true"
            displayBioSamples="true"
            experimentDisplayType="Quantitative Value"
            quantitativeDisplayType="Top 3 Precursor Intensity"
            path="."/>
```

experiment-report and **isoform-report** have these additional options, using:

experimentDisplayType="Quantitative Value"

a) quantitativeDisplayType:

```
Total Spectra
Average TIC
Total TIC
Top 3 TIC
Average Precursor Intensity
Total Precursor Intensity
Top 3 Precursor Intensity
emPAI
Weighted Spectra
NSAF
```

Note that only one Quantitative Value export can be specified per SCAFML. Multiple **<Export/>** elements with this attribute will give incorrect results.

The default value of useNormalization (if the attribute is not given) is true (use normalization).

If the user selects NSAF (which does require normalization) as display type, the required setting will be used.

If the user specifies the wrong kind of normalization for the given display type, an error message is written and the required normalization is used.

```
       <Annotation>
           <GOA location="/path/to/goa.db"/>
       </Annotation>
```

Annotations option: Gene Ontology annotations to the Annotation element.

If the following GOA element is specified in the **Annotation** element, this will supersede any value given for the attribute **annotateWithGOA** attribute in the **Experiment** element. We will write to the output log: ***"GO annotations applied to: x proteins using db: y"***, where x is the number of proteins annotated and y the absolute path to the goa db. We expect the location given is a valid ***.db** file (eg, configured using Scaffold's GUI).

```
       <Annotation id="star">
           <Star color="Orange and Blue"
               delimiter="\s+" regEx="true">
                   \bALBU_BOVIN\b
```

In this case, **id** is **"star"**, and the delimiter is regex "at least one white space".

The **"\b"** is regex whole word match, eg, **ALBU_BOVIN** or **ALBU_HUMAN**

```
           <Star color="Blue"
               delimiter="\s+" regEx="true">

                   (\bALBU_BOVIN\b)|(\bALBU_HUMAN\b)
```

For multiple different star designations, add multiple Star elements inside the Annotation element. To match more than one accession, use the regex for "or" (eg, the pipe character: **|**).

```
           </Star>
           <hide delimiter=",">
               keratin,albumin
           </hide>
```

Other option is **"hide"**, where **id** is **"hide"**, with delimiter value specified.

You can specify color as an attribute for the **Star** element with values: **"Empty", "Orange", "Blue",** or **"Orange and Blue".**

```
        </Annotation>
    </Experiment>
</Scaffold>
```

Close the **Annotation** element, before closing **Experiment** element.

The **Scaffold** element is the parent element for the document.

Appendix A

Running Scaffold Batch with the –h option gives you an example driver file, and lists export details. For example:

```
Here are the following attributes considered for a particular `type` of export: An attribute's
default value is used if the attribute is not provided. Details for each attribute are
provided in a table at the end.

  type: sfd; sf3; scaffold
    - saveOnlyIdentifiedSpectra
    - saveNoSpectra
    - discardBelowThresholds
    - discardSpectraWithPeptidesCount
    - discardUnlessModifiedWith
    - saveFrozen
    - saveCondensed
    - suffix
    - thresholds
    - timestamp
  type: scaffoldxml
    - suffix
    - thresholds
    - timestamp
  type: mzidentml; mzid
    - version
    - showDecoys
    - useFilter
    - individualReports
    - useGzip
    - writePeaklists
    - showHiddenProteins
    - suffix
    - thresholds
    - timestamp
  type: protxml
    - suffix
    - thresholds
    - timestamp
  type: spectrum
    - exportDiscriminantScores
    - suffix
    - thresholds
    - timestamp
  type: statistics
    - suffix
    - thresholds
    - timestamp
```

```
type: peptide-report
  - useQPlusReport
  - suffix
  - thresholds
  - timestamp
type: spectrum-report
  - useQPlusReport
  - suffix
  - thresholds
  - timestamp
type: protein-report
  - suffix
  - thresholds
  - timestamp
type: accession-report
  - suffix
  - thresholds
  - timestamp
type: publication-report
  - useQPlusReport
  - suffix
  - thresholds
  - timestamp
type: experiment-report
  - experimentDisplayType
  - displayBioSamples
  - includeFamilies
  - suffix
  - thresholds
  - timestamp
type: isoform-report
  - experimentDisplayType
  - displayBioSamples
  - includeFamilies
  - suffix
  - thresholds
  - timestamp
type: spectrum-counting-report
  - includeFamilies
  - suffix
  - thresholds
  - timestamp
type: sqml
  - suffix
```

```
    - thresholds
    - timestamp
  type: protein-quantitation-xml
    - suffix
    - thresholds
    - timestamp
  type: samples-report
    - reportIsoforms
    - includeFamilies
    - displayBioSamples
    - suffix
    - thresholds
    - timestamp
  type: raw-data-report
    - suffix
    - thresholds
    - timestamp
  type: statistical-analysis-report
    - multiplexQuantSettingsId
    - suffix
    - thresholds
    - timestamp
```

Appendix B

Truth table for grouping and clustering elements and attributes options.

We still maintain backward compatibility with legacy attributes. The truth table below shows the combinations of the legacy attributes and their respective behavior. For example, using **`proteinGrouping=experiment-wide`** is the same as if you had the attributes **`useFamilyProteinGrouping=false`** and **`useIndependentSampleGrouping=false`**

	useFamilyProteinGrouping="true"	useFamilyProteinGrouping="false"
useIndependentSampleGrouping="true"	protein-cluster-analysis	independent-samples
useIndependentSampleGrouping="false"	protein-cluster-analysis	experiment-wide

If none of the attributes: **`proteinGrouping`**, **`useFamilyProteinGrouping`** or **`useIndependentSampleGrouping`** exist, **`protein-cluster-analysis`** will be used.