Advanced Configuration of ProteinPilot™ Software

RUO-IDV-06-1665-C

September 2014

AB Sciex Pte. Ltd.
Blk 33, #04-06
Marsiling Ind Estate Road 3
Woodlands Central Indus. Estate
SINGAPORE 739256

Contents

Introduction

This document explains the concept of parameter translation used by the Paragon™ algorithm in the ProteinPilot™ software. The document assumes a basic knowledge of XML and provides instructions on how to customize the ProteinPilot software.

Since ProteinPilot 3.0 software, it is possible to modify many parameter settings that were previously hidden. These settings are in the **ParameterTranslation.xml** file. This file controls the translation of the lab-centric options that appear in the Paragon Method dialog to the much more complex parameters used by the Paragon algorithm.

Users can customize a wide range of parameters, including modification sets, digestion sets, and instrument settings, as well as their appearance in the user interface. By controlling the parameter translation, the ProteinPilot software can be customized for the workflows that are most important to each lab.

Users are encouraged to share improvements and additions to workflows. Refer to How to Share Improvements with Others on page 47.

About the Paragon™ Algorithm

The following information about the Paragon algorithm is intended to help users better understand the customization described in this document.

The Paragon algorithm performs two types of searches. Both components score peptide hypotheses in the same way, essentially counting up the number of matching b and y ions, but the two components differ regarding the peptide hypothesis selected for scoring.

- **Fraglet**: similar to a conventional database search.

  Peptide hypotheses are selected solely on the basis of precursor mass, that is, matching the experimental precursor mass to the mass of the peptide hypothesis mass within a certain tolerance. MS/MS information is used in the scoring process but is not used in the selection process. Only peptide hypotheses with feature probabilities above a specific threshold are scored. Fraglet performance degrades when the search tries to account for too many features.

- **Taglet**: the unique aspect of the Paragon algorithm.

  Peptide hypotheses are selected on the basis of sequence tags, derived from the MS/MS peaks, together with feature probabilities. All features, based on the settings in the Paragon Method dialog, are considered. The highest probability features are considered for all regions of the database, while the lowest probability features are considered only for regions that are

very strongly implicated by tag evidence. Modifications that are not possible, such as a modification resulting from a cysteine alkylation other than the one used for sample preparation, are never considered. Refer to Setting Feature Probabilities on page 42.

Users can also refer to Shilov, I. V. et al., *Mol. Cell. Proteomics* 6, 1638-1655 (2007) (http://www.mcponline.org/cgi/content/full/6/9/1638) for further information about Taglet.

The **Search Effort** setting in the **Paragon Method** dialog determines which searches are performed:

- **Rapid**: only a Fraglet search is performed.
- **Thorough**: both Fraglet and Taglet searches are performed, except when Digestion is set to **None** in the **Paragon Method** dialog. In that case, only Taglet searches are performed.

After the searches are complete, the Pro Group algorithm assembles the results into a minimal list of detected proteins.

The Parameter Translation Concept

Parameter translation is the process by which knowledge of reagents and actions in the lab are translated into their complex informatics implications.

Figure 1 shows how the Parameter Translation file defines both the choices available in the **Paragon Method** dialog and how the system reacts to the selection of these options.

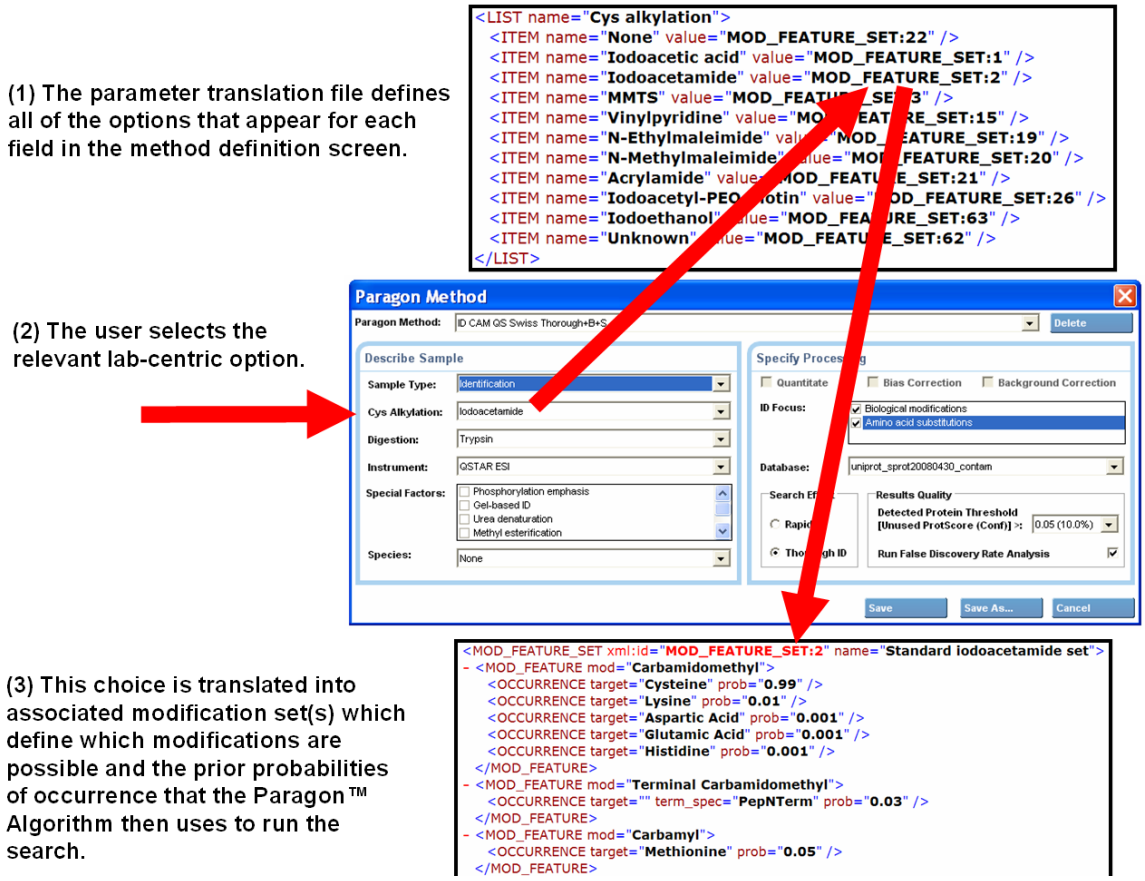

Figure 1 – Example of the role of the Parameter Translation file

Parameter translation in the ProteinPilot software serves several purposes:

- To help users select search parameters successfully, even with limited informatics expertise.
- To group a large number of algorithm-centric parameter decisions into a smaller number of user interface settings.
- To provide the ProteinPilot software with feature probabilities. These probabilities enable the software to make certain decisions without user input.
    - For Fraglet, the search parameters are based on only the most common features. Similar to other conventional search engines, the performance of Fraglet degrades when the search tries to account for too many rare features.
    - For Taglet, all possible features are searched for. Higher probability features are searched for in more regions of the database than lower probability features. Some features can be ruled out due to the workflow and are not searched for. For example, modifications resulting from any cysteine alkylation reagent other than the specified one can be ruled out.

    Refer to Setting Feature Probabilities on page 42.

About the Basic Architecture

Two files are involved with the parameter translation process:

- **ProteinPilot.DataDictionary.xml**
  This is the ProteinPilot software Data Dictionary, referred to as the Data Dictionary in this document.

  This file defines fundamental chemical building blocks, such as elements, amino acids, and modifications. Its schema is described in Overview of the Data Dictionary File Schema on page 9. The file is located in the following folder:

  `C:\Program Files\AB SCIEX\ProteinPilot`

- **ParameterTranslation.xml**
  This is referred to as the Parameter Translation file in this document.

  This file contains workflow parameter sets that describe modifications, substitutions, digestion agents, instrument settings, quantitation schemes, and more, as well as information about how these workflow parameter sets relate to what is shown in the **Paragon Method** dialog. Its schema is described in Overview of the Parameter Translation File Schema on page 9. The file is located in the following folder:

  `C:\Program Files\AB SCIEX\ProteinPilot\WorkflowDirectory`

NOTE*:* Modifications made to the ParameterTranslation.xml file can be used only in the same version of the ProteinPilot software in which they are created. For example, v. 5.0 .xml files can be used only in ProteinPilot 5.0 software.

OVERVIEW OF THE DATA DICTIONARY FILE SCHEMA

The Data Dictionary describes physical constants. It contains definitions of atomic elements, amino acids, modifications, cleavage agents, and adducts. The atomic elements are used in the chemical formulas which define the amino acids, modifications, and adduct ions. The Data Dictionary schema has the following structure:

```
<DATADICTIONARY>
    <El...(defnition of one element and its mass, including isotopes)>
    <El...(defnition of one element and its mass, including isotopes)>
    …..
    <AA…(definition of one amino acid)>
    <AA…(definition of one amino acid)>
    …..
    <MOD…(definition of one modification)>
    <MOD…(definition of one modification)>
    ….
    <BMd..(definition of disulfide bridge modification)>
    <CAgt…(definition of one cleavage agent)>
    <CAgt…(definition of one cleavage agent)>
    …
    <AIn…(definition of one adduct ion)>
    <AIn…(definition of one adduct ion)>
    …
</DATADICTIONARY>
```

OVERVIEW OF THE PARAMETER TRANSLATION FILE SCHEMA

The Parameter Translation file schema has the following structure:

```
<TRANSLATIONS>
    <USER_INPUT_TRANSLATIONS>
        <LIST…(list of items for one user interface control in the Paragon Method dialog)>
            <ITEM…(an item in the list)>
            <ITEM…(an item in the list)>
            <ITEM…(an item in the list)>
        </LIST>
        <LIST…(list of items for one user interface control in the Paragon Method dialog)>
        <LIST…(list of items for one user interface control in the Paragon Method dialog)>
    </USER INPUT TRANSLATIONS>

    <QUANT_TYPE…(defines a label-based quantitation scheme)>
```

```
    <QUANT_TYPE…(defines a label-based quantitation scheme)>
    <QUANT_TYPE…(defines a label-based quantitation scheme)>

<SPECIES_SPECIES_MATRIX>
        <INTERSECTION…(maps one species in a FASTA file to a species in ProteinPilot Software)>
        <INTERSECTION…(maps one species in a FASTA file to a species in ProteinPilot Software)>
        <INTERSECTION…(maps one species in a FASTA file to a species in ProteinPilot Software)>
    </SPECIES_SPECIES_MATRIX>

    <MOD_FEATURE_SET…(defines a set of modifications)>
        <MOD_FEATURE…(the name and probability of one type of modification in the set)>
        <MOD_FEATURE…(the name and probability of one type of modification in the set)>
        <MOD_FEATURE…(the name and probability of one type of modification in the set)>
    </MOD_FEATURE_SET>
    <MOD_FEATURE_SET…(defines a set of modifications)>
    <MOD_FEATURE_SET…(defines a set of modifications)>
    …

    <SUBSTITUTION_SET… (describes the probabilities for substituting one amino acid for another)>

    <DIGEST_SET… (defines a digestion set)>
    <DIGEST_SET… (defines a digestion set)>
    <DIGEST_SET… (defines a digestion set)>
    …
    <INSTRUMENT…(parameters for how the Paragon algorithm treats data for an instrument type)>
    <INSTRUMENT…(parameters for how the Paragon algorithm treats data for an instrument type)>
    <INSTRUMENT…(parameters for how the Paragon algorithm treats data for an instrument type)>
    …
</TRANSLATIONS>
```

Working with the Parameter Translation File

Whether to Create a New Set or Modify an Existing Set

In general, it is recommended that users create new workflow parameter sets (for example, modification sets, digestion sets, quantitation schemes) instead of editing existing workflow parameter sets. Users can copy an existing set and then edit it to address individual needs.

How to Recover from an Error

If errors are made (for example, unwanted changes), delete the current **ParameterTranslation.xml file** or **ProteinPilot.DataDictionary.xml file**, and then restart the ProteinPilot software. When the software restarts, it recognizes that the file is missing and then regenerates the original.

Workflow Support Might Be Hidden

In ProteinPilot 3.0 software, some experimental workflows are present in the Parameter Translation file but they do not appear in the **Paragon Method** dialog. An ITEM that is considered experimental has the *method_filter* attribute with a value that includes the word EXPERIMENTAL.

These experimental workflows have not been validated and are not part of the supported ProteinPilot software.

Since ProteinPilot 4.0 software, no items are hidden using this mechanism. Instead, the *Paragon™ Method Settings* document in the Help folder provides a description of how well tested and optimized each feature setting is.

The EXPERIMENTAL *method_filter* can hide items from the user interface without deleting them and also allows users to comment items out.

In the example below, the items in red are experimental.

```
<LIST name="Digestion">
    <ITEM name="Trypsin" value="DIGEST_SET:1"/>
    <ITEM name="Glu C" value="DIGEST_SET:4" />
    <ITEM name="Chymotrypsin" value="DIGEST_SET:2" />
    <ITEM name="CNBr" value="DIGEST_SET:5" />
    <ITEM name="Lys C" value="DIGEST_SET:6" />
    <ITEM name="Acid Cleavage" value="DIGEST_SET:7" />
    <ITEM name="Arg C" value="DIGEST_SET:8" />
    <ITEM name="Asp N" value="DIGEST_SET:9" />
    <ITEM name="Trypsin + Chymotrypsin" value="DIGEST_SET:10"
method_filter="ABSCX.PROPILOT.EXPERIMENTAL,2.0"/>
    <ITEM name="Trypsin + Glu C" value="DIGEST_SET:11"
method_filter="ABSCX.PROPILOT.EXPERIMENTAL,2.0"/>
    <ITEM name="Trypsin + Asp N" value="DIGEST_SET:12"
method_filter="ABSCX.PROPILOT.EXPERIMENTAL,2.0"/>
    <ITEM name="Lys C + Chymotrypsin" value="DIGEST_SET:13"
method_filter="ABSCX.PROPILOT.EXPERIMENTAL,2.0"/>
    <ITEM name="Lys C + Glu C" value="DIGEST_SET:14"
method_filter="ABSCX.PROPILOT.EXPERIMENTAL,2.0"/>
    <ITEM name="None" value="DIGEST_SET:3"/>
</LIST>
```

To show a hidden workflow, delete both the attribute name and the value for the *method_filter* attribute.

Implementing a New Workflow

Implement a new workflow

1. Create a new workflow parameter set or modify an existing workflow parameter set.

   The remainder of this document provides instructions for each type of workflow parameter set.

2. If the workflow parameter set is new, then connect the user interface to the newly created workflow parameter set.
   If an existing workflow parameter set has been modified, then go to step 3.

3. Save and close all of the files, and then restart the ProteinPilot software.

4. Test the new workflow:
    - Verify that any new user interface options appear in the **Paragon Method** dialog.
    - Verify that new modifications are found in searches as expected.
    - Verify that the results are reasonable, as measured by the false discovery rate analysis.

Connecting a Workflow Parameter Set to the User Interface

The USER_INPUT_TRANSLATIONS element links all of the workflow parameter sets to the user interface.

A section of the USER_INPUT_TRANSLATIONS element follows. Each LIST element represents a list or a check box in the **Paragon Method** dialog (except for the WORKFLOWS element).

Each LIST element contains one or more ITEM elements, each of which corresponds to an option for the LIST element.

The expanded section of the USER_INPUT_TRANSLATIONS element shows part of the **Digestion** list in the **Paragon Method** dialog. Users can select from Trypsin, Glu C, Chymotrypsin, and additional options on the **Digestion** list. If Trypsin is selected in the **Paragon Method** dialog, then the search uses the workflow parameter set DIGEST_SET:1.

```
<USER_INPUT_TRANSLATIONS>
    <LIST name=”Workflows”>
    <LIST name="Sample type">
    <LIST name=”Species”>
    <LIST name=”Cys alkylation”>
    <LIST name="Digestion">
        <ITEM name="Trypsin" value="DIGEST_SET:1"/>
        <ITEM name="Glu C" value="DIGEST_SET:4" />
        <ITEM name="Chymotrypsin" value="DIGEST_SET:2" />
        <ITEM name="CNBr" value="DIGEST_SET:5" />
        …
    </LIST>
    <LIST name=”Special factors”>
    <LIST name=”Identification focus”>
    <LIST name=”Instrument”>
</USER_INPUT_TRANSLATIONS>
```

The following table describes the attributes of the ITEM element.

Attributes of the ITEM Element	
Attribute	**Description**
Name	Specifies the text shown in the lists and check boxes in the Paragon Method dialog. *Name* is the only required attribute.
value	The xml:id of the workflow parameter sets associated with the ITEM. The allowed parameter sets depend on the LIST item: ● Modification set: defined by a MOD_FEATURE_SET element. Users can specify multiple modification sets, separated by commas. ● Substitution set: defined by a SUBSTITUTION_SET element. ● Digestion set: defined by a DIGEST_SET element. Only one digestion set is allowed per ITEM. To specify multiple digestion agents, users must summarize their aggregate behavior as one digestion set. ● Instrument definition: defined by an INSTRUMENT element. Only one instrument definition is allowed per item. If users are mixing data from multiple instruments, then create one INSTRUMENT definition that balances the characteristics of the data.
value1	When present, specifies the xml:id for the associated quantitation workflow parameter set. A quantitation workflow parameter set is defined by the QUANT_TYPE element. Only one *value1* attribute is allowed per ITEM, and the *value1* attribute is used only by the Sample type list.
method_filter	Controls the display of the ITEM in the user interface. Refer to Workflow Support Might Be Hidden on page 11.

Connect a new workflow parameter set to the user interface

1. Note the xml:id of the new workflow parameter set.
2. Determine the LIST element to which the workflow parameter set should belong.
3. Add a new ITEM to the appropriate LIST element, with the xml:id as the *value* attribute, or edit an existing ITEM to use the xml:id in the *value* attribute.
4. If more than one ITEM element needs to be added or modified, repeat steps 2 and 3.

Adding a New Modification Set

Adding a Modification Set – General Steps

Modifications are described in two places. The Data Dictionary describes the chemical formula as well as how the modification interacts with amino acids, while the Parameter Translation file describes the feature probabilities for various scenarios.

Add a new modification set

Prerequisite

Confirm that the required modifications are already defined in the Data Dictionary. If necessary, add the modification to the Data Dictionary.

The ProteinPilot software uses the MS synonym subset of the PSI-MOD standard (*Nature Biotechnology* (2008) 26, 864 – 866 http://www.nature.com/nbt/journal/v26/n8/full/nbt0808-864.html). These names can be found at http://unimod.org, in the PSI-Mod Name field for a modification entry.

1. Create a new MOD_FEATURE_SET element with a unique xml:id.

   NOTE: Optionally, users can copy a similar, existing MOD_FEATURE_SET element, and then edit the copy.

2. Connect the user interface to the new MOD_FEATURE_SET element.

   Refer to Connecting a Workflow Parameter Set to the User Interface on page 12.

3. Save and close all of the files, and then restart the ProteinPilot software.

Adding a Modification Set – An Example

The following example shows how to create a simpler version of the existing gel-based ID special factor set.

The following is an example of an existing modification. In this example, all of the modifications in the new modification set already exist in the Data Dictionary, so no changes to the Data Dictionary are required.

```
<Mod rKey="0">
    <Nme>Oxidation</Nme>

    <UniModAcc>35</UniModAcc>
    <TLC>1Ox</TLC>
    <TS>255</TS>
    <Fma>O</Fma>
    <RpF></RpF>
    <Tgt>Cysteine</Tgt>
```

```
        <Tgt>Aspartic Acid</Tgt>
        <Tgt>Phenylalanine</Tgt>
        <Tgt>Histidine</Tgt>
        <Tgt>Lysine</Tgt>
        <Tgt>Methionine</Tgt>
        <Tgt>Asparagine</Tgt>
        <Tgt>Proline</Tgt>
        <Tgt>Arginine</Tgt>
        <Tgt>Selenocysteine</Tgt>
        <Tgt>Tryptophan</Tgt>
        <Tgt>Tyrosine</Tgt>
        <NLF></NLF>
        <IIF></IIF>
        <Chg>0</Chg>
    </Mod>
```

The following table describes the elements and attributes of the Mod element in the ProteinPilot.DataDictionary.xml file.

Elements and Attributes in of the Mod Element in the Data Dictionary

Element or Attribute	Description
Mod	Describes one modification. If there are different specificities for the same modification, define each as a separate Mod element. For example, acetylation must be defined separately to allow it on lysine versus limiting it to peptide N-termini versus limiting it to only protein N-termini. Refer to the Acetyl, Terminal Acetyl, and Protein Terminal Acetyl modifications in the Data Dictionary for examples.
Nme	The modification name. This is generally based on the PSI-Mod nomenclature, found at http://unimod.org. The only exception is for specificity variant modifications, which are extensions of the standard.
UniModAcc	UniMod Accession #, if the modification maps to a UniMod mod.
TLC	The three-letter code for the modification, used to indicate the modification in the peptide sequence field of the Fragmentation Evidence for Peptide pane in the ProteinPilot software. This code does not need to be unique. Terminal-specificity variant modifications can use the same three-letter code. For example, the three acetyl variations mentioned in the Mod element definition have “1AC” as their three letter code.
Tgt	A residue that might be a “target” (a place where this modification might occur). The residue must be defined in an AA element in the Data Dictionary. When no target residues are defined, all residues are allowed.

Elements and Attributes in of the Mod Element in the Data Dictionary	
Element or Attribute	**Description**
TS	Defines the terminal specificity of the modification. Allowed values are: ● 255 – No terminal specificity ● 5 – Only any peptide N terminus ● 4 – Only the protein N terminus ● 8 – Only the protein C terminus ● 10 – Only any peptide C terminus
RpF	The moiety that is replaced by the modification (replaced formula). This is optional and required only when elements are lost. Refer to *Fma*.
Fma	Either the entire molecular formula or, if the RpF element is used, the formula for the added elements. Since the moiety replaced by a modification is often hydrogen and the modification itself often contains hydrogen, the net result of the modification can be indicated entirely by the *Fma* element without the *RpF* element. For example, methylation can be indicated as either: ● Fma=CH3 and RpF=H ● Fma=CH2 and a blank RpF element The letters used in these formulae must be defined in the Data Dictionary, including definitions of isotopic variant modifications. Refer to <Nme>iTRAQ4plex</Nme> in the Data Dictionary for an example, as well as the use elements *Cb* and *Nc*, which is how C-13 and N-15 are defined in the element section of the Data Dictionary.
NLF	Neutral loss formula.

Elements and Attributes in of the Mod Element in the Data Dictionary	
Element or Attribute	**Description**
DisplayName	(not shown in the example) The name shown in the **Modifications** column in the ProteinPilot software results. If *DisplayName* is not specified, *Nme* is shown. This element allows terminal-specific modifications to use the same name as a non-terminal-specific occurrence of the same modification. For example, consider Carboxymethyl versus Terminal Carboxymethyl or Acetyl versus Terminal Acetyl. DisplayName can also be used to group variants of a common modification moiety for other reasons. Consider the following phosphorylation variants: ● <Nme>Phospho (Ser,Thr)</Nme>, which allows for neutral loss ● <Nme>Phospho</Nme>, which does not Because the *DisplayName* and *TLC* for <Nme>Phospho (Ser,Thr)</Nme> are the same as *Nme* and *TLC* for <Nme>Phospho</Nme>, both variants appear as “Phospho” in the results.

Because all of the modifications are already present in the Data Dictionary, the next step is to add a new modification feature set for the simpler gel-based ID set to the Parameter Translation file.

A modification feature set consists of one or more MOD_FEATURE elements, each of which uses a modification from the Data Dictionary and specifies the probability of that modification on particular target amino acids.

In the MOD_FEATURE element, both the name of the modification (the *mod* attribute) and the target amino acids (the *target* attribute) must match the Data Dictionary.

```
<MOD_FEATURE_SET xml:id="MOD_FEATURE_SET:41" name="Phosphorylation emphasis">

    <MOD_FEATURE mod="Dehydrated" >
        <OCCURRENCE target="Serine" prob="0.02" mod_class="biomod" />
        <OCCURRENCE target="Threonine" prob="0.02" mod_class="biomod" />
    </MOD_FEATURE>

    <MOD_FEATURE mod="Phospho" >
        <OCCURRENCE target="Cysteine" prob="0.01" mod_class="biomod" />
        <OCCURRENCE target="Aspartic Acid" prob="0.01" mod_class="biomod" />
        <OCCURRENCE target="Histidine" prob="0.01" mod_class="biomod" />
        <OCCURRENCE target="Lysine" prob="0.01" mod_class="biomod" />
        <OCCURRENCE target="Arginine" prob="0.01" mod_class="biomod" />
```

```
        <OCCURRENCE target="Tyrosine" prob="0.05" mod_class="biomod" />
    </MOD_FEATURE>

    <MOD_FEATURE mod="Phospho(Ser,Thr)" >
        <OCCURRENCE target="Serine" prob="0.35" mod_class="biomod" />
        <OCCURRENCE target="Threonine" prob="0.35" mod_class="biomod" />
    </MOD_FEATURE>

</MOD_FEATURE_SET>
```

The following table describes the elements and attributes of the MOD_FEATURE element in the ParameterTranslation.xml file.

Elements and Attributes for the MOD_FEATURE Element in the Parameter Translation File

Element or Attribute	Description
MOD_FEATURE	One element per modification name, with a one-to-one correspondence to the Data Dictionary. The separate treatment of different terminal specificities also applies here.
Mod	The *Nme* element for the modification, from the Data Dictionary.
OCCURRENCE	One possible location for the modification and the probability of this modification at that location.
Target	The amino acid residue that can be modified. Values for target must match the names in the Data Dictionary.
mod_class	Used to classify a modification type. A BIOMOD value indicates a modification is a biological feature and will be listed in the Features Tab when detected.
term_spec	Indicates terminal specificity of the modification, if any. Allowed values are: ● PepCTerm – specific to peptide C-terminal ● PepNTerm – specific to peptide N-terminal ● ProtCTerm – specific to protein C-terminal ● ProtNTerm – specific to protein N-terminal
prob	The prior probability of the occurrence of this modification on this residue, per instance of the residue. For example, if *target*=Methionine and *prob*=0.20, the expectation is that 20% of methionine residues will be oxidized.

NOTE: The Paragon algorithm treats non-terminal, peptide-terminal-specific, and protein-terminal-specific versions of any modification as separate modifications. Each version of the modification must be a separate entry in the Data Dictionary, and the Parameter Translation file must respect this separation.

In this example, “Propionamide” and “Terminal Propionamide” are separate. Although it might seem logical, this cannot be done:

```
<MOD_FEATURE mod="Propionamide">
    <OCCURRENCE target="Cysteine" prob="0.10"/>
    <OCCURRENCE target="Lysine" prob="0.01"/>
    <OCCURRENCE target=" " term_spec="PepNTerm" prob="0.01"/>
</MOD_FEATURE>
```

Instead, two MOD_FEATURE elements must be used, as shown in the MOD_FEATURE_SET.

A Closer Look at Oxidation in the Parameter Translation File

The oxidation modification is found in several places in the Parameter Translation file. The standard workup modification set establishes default probabilities for the occurrence of oxidation on several residues:

```
<MOD_FEATURE mod="Oxidation">
    <OCCURRENCE target="Methionine" prob="0.15"/>
    <OCCURRENCE target="Cysteine" prob="0.00001"/>
    <OCCURRENCE target="Histidine" prob="0.005"/>
    <OCCURRENCE target="Tryptophan" prob="0.013"/>
    <OCCURRENCE target="Proline" prob="0.01"/>
</MOD_FEATURE>
```

Oxidation is also included in the biological modification set, but with different OCCURRENCE elements:

```
<MOD_FEATURE mod="Oxidation" >
            <OCCURRENCE target="Cysteine" prob="0.001" mod_class="biomod" />
            <OCCURRENCE target="Aspartic Acid" prob="0.0009" mod_class="biomod" />
            <OCCURRENCE target="Phenylalanine" prob="0.001" mod_class="biomod" />
            <OCCURRENCE target="Lysine" prob="0.00005" mod_class="biomod" />
            <OCCURRENCE target="Asparagine" prob="0.0012" mod_class="biomod" />
            <OCCURRENCE target="Proline" prob="0.01" mod_class="biomod" />
            <OCCURRENCE target="Arginine" prob="0.0014" mod_class="biomod" />
            <OCCURRENCE target="Selenocysteine" prob="0.00001" mod_class="biomod" />
            <OCCURRENCE target="Tyrosine" prob="0.0013" mod_class="biomod" />
        </MOD_FEATURE>
```

If the **Biological modifications** check box is selected in the Paragon Method dialog, then the Paragon algorithm searches for oxidation on more residues.

Oxidation is also included in the modification set for the Gel-Based ID option in the **Special Factors** list:

```
<MOD_FEATURE mod="Oxidation">
    <OCCURRENCE target="Methionine" prob="0.4"/>
    <OCCURRENCE target="Tryptophan" prob="0.013"/>
    <OCCURRENCE target="Cysteine" prob="0.001"/>
    <OCCURRENCE target="Histidine" prob="0.003"/>
    <OCCURRENCE target="Proline" prob="0.03"/>
</MOD_FEATURE>
```

If the Gel-based ID option is selected in the **Paragon Method** dialog, the Paragon algorithm searches for oxidation more extensively, due to the higher feature probabilities. This gives the Paragon algorithm the chemical intelligence that oxidation artifact modifications are more likely to be found when the data comes from a gel-based workflow. Refer to Modification Probabilities on page 42.

Adding a New Label-Based Quantitation Scheme

The ProteinPilot software supports three types of quantitation analyses:

- **MS/MS-based isobaric quantitation using the SCIEX iTRAQ® reagents**: This includes the 4plex and 8plex variants.
- **MS-based duplex, with both heavy and light modified**: For example, all variants of the ICAT reagents. Both the light and the heavy forms are labeled with a variant of the reagents. The Paragon algorithm performs two separate searches, one considering only the light form and the other considering the heavy form, preventing heavy and light modifications from being found on the same peptide. The SCIEX mTRAQ™ reagents are also in this category.
- **MS-based duplex, with the light form unmodified**: The most common example is SILAC. The light form is simply the unaltered version of the target amino acids. The Paragon algorithm performs a single search for this type of quantitation. Heavy oxygen derivatives are also in this category.

When adding a new quantitation scheme, it can be useful to copy the elements of an existing method. Choose the method closest to the new method to be added. Model the new quantitation scheme as a derivative of an existing method from one of the categories. For example, to add duplex ICPL support, create a derivative of the ICAT method.

Adding a Quantitation Scheme – Guidelines

- **Limited quantitation support for other vendor instruments**: All types of quantitation analyses enabled by the ProteinPilot software are supported for input of AB SCIEX native data. Only MS/MS-based isobaric quantitation using the SCIEX iTRAQ® reagents is supported when starting from .mgf files.
- **Mass delta limitations for MS-based quantitation**: Although an MS-based quantitation method for a duplex that differs by any mass delta can be created, it is recommended that a mass delta of at least 4 Da, and preferably 6 Da, is used. This is because the MS-based quantitation method does not perform any deconvolution of overlapping isotope series. If the isotope series from the light labeled peptide extends into the heavy peptide's isotope series, the quantitation results are inaccurate. This is true only of .wiff data. For TOF/TOF™ Series Explorer™ software data, the data

is already deconvolved when it is extracted from the database. In this case, it is not an issue that the quantitation algorithm does not do this.

- **Co-elution of heavy and light forms is required**: In some labeling schemes, the heavy and light form peptides do not co-elute from the column. This is generally the case when deuterium is used in labeling, as for the first generation of ICAT reagents. In the Cleavable ICAT reagents, C-13 is used to create the mass shift, yielding perfectly co-eluting heavy and light forms. The quantitation algorithm requires that the reagents co-elute because the results are derived from a single spectrum. Any non-co-eluting labeling methods will yield biased results.
- **Alternate isobaric reagents are not supported**: The software is hard-coded to support SCIEX iTRAQ® reagents.

Adding a Quantitation Scheme: Overview

Adding a new quantitation scheme is very similar to adding a new modification set.

Add a new quantitation scheme

1. If necessary, create any new MOD_FEATURE_SET elements required for the label. Refer to Adding a New Modification Set on page 15.

   For MS-based duplex, where heavy and light are both modified, two separate MOD_FEATURE_SET elements are needed for the heavy and light variants of the label.

2. Create a new quantitation scheme by creating a new QUANT_TYPE element with a unique xml:id.

   NOTE: Users might want to copy a similar, existing QUANT_TYPE element, and then edit the copy.

3. Connect the user interface to the new MOD_FEATURE_SET and QUANT_TYPE elements by editing the Sample type list.

   Refer to Connecting a Workflow Parameter Set to the User Interface on page 12.

4. Save and close all of the files, and then restart the ProteinPilot software.

Adding a Quantitation Scheme: Details

The **Sample Type** list in the USER_INPUT_TRANSLATIONS element shows a list similar to the following:

```
<LIST name="Sample type">
    <ITEM name="Identification" value="MOD_FEATURE_SET:1"/>
    <ITEM name="iTRAQ 8plex (Peptide Labeled)"
    value="MOD_FEATURE_SET:73,MOD_FEATURE_SET:1" value1="iTRAQ8PLEX"/>
    <ITEM name="iTRAQ 8plex (Protein Labeled)"
    value="MOD_FEATURE_SET:74,MOD_FEATURE_SET:1" value1="iTRAQ8PLEX"/>
    <ITEM name="iTRAQ 4plex (Peptide Labeled)"
    value="MOD_FEATURE_SET:71,MOD_FEATURE_SET:1" value1="iTRAQ4PLEX"/>
    <ITEM name="iTRAQ 4plex (Protein Labeled)"
    value="MOD_FEATURE_SET:72,MOD_FEATURE_SET:1" value1="iTRAQ4PLEX"/>
    <ITEM name="mTRAQ (Peptide Labeled - M00, M04)" value="MOD_FEATURE_SET:1"
    value1="mTRAQ_0-4"/>
    <ITEM name="mTRAQ (Peptide Labeled - M00, M08)" value="MOD_FEATURE_SET:1"
    value1="mTRAQ_0-8"/>
    <ITEM name="mTRAQ (Peptide Labeled - M04, M08)" value="MOD_FEATURE_SET:1"
    value1="mTRAQ_4-8"/>
    <ITEM name="mTRAQ (Peptide Labeled - M00, M04, M08 - ID only)"
    value="MOD_FEATURE_SET:1,MOD_FEATURE_SET:108"/>
    <ITEM name="SILAC (Lys+6, Arg+10)" value="MOD_FEATURE_SET:1" value1="SILAC (Lys+6,
    Arg+10)"/>
    <ITEM name="SILAC (Lys+6)" value="MOD_FEATURE_SET:1" value1="SILAC (Lys+6)"/>
    <ITEM name="SILAC (Lys+8)" value="MOD_FEATURE_SET:1" value1="SILAC (Lys+8)"/>
    <ITEM name="SILAC (Arg+10)" value="MOD_FEATURE_SET:1" value1="SILAC (Arg+10)"/>
    <ITEM name="SILAC (Lys+6, Arg+6)" value="MOD_FEATURE_SET:1" value1="SILAC (Lys+6,
    Arg+6)"/>
    <ITEM name="SILAC (Lys+8, Arg+10)" value="MOD_FEATURE_SET:1" value1="SILAC (Lys+8,
    Arg+10)"/>
    <ITEM name="SILAC (Arg+6)" value="MOD_FEATURE_SET:1" value1="SILAC (Arg+6)"/>
    <ITEM name="SILAC (Arg+4)" value="MOD_FEATURE_SET:1" value1="SILAC (Arg+4)"/>
    <ITEM name="SILAC (Ile+6)" value="MOD_FEATURE_SET:1" value1="SILAC (Ile+6)"/>
    <ITEM name="Proteolytic O-18 labeling" value="MOD_FEATURE_SET:1" value1="Proteolytic O-18 v
    O-16"/>
    <ITEM name="Cleavable ICAT" value="MOD_FEATURE_SET:1" value1="ICAT9"/>
    <!-- <ITEM name="Deuterium ICAT" value="MOD_FEATURE_SET:1" value1="ICAT8"/> -->
    <ITEM name="ICPL Light, Heavy (Peptide Labeled)" value="MOD_FEATURE_SET:1" value1="ICPL
    peptide" />
    <ITEM name="ICPL Light, Heavy (Protein Labeled)" value="MOD_FEATURE_SET:1" value1="ICPL
    protein" />
    <ITEM name="iTRAQ 8plex (Peptide Labeled) w other K and N-term mods possible"
    value="MOD_FEATURE_SET:76,MOD_FEATURE_SET:1" value1="iTRAQ8PLEX"/>
    <ITEM name="iTRAQ 4plex (Peptide Labeled) w other K and N-term mods possible"
    value="MOD_FEATURE_SET:75,MOD_FEATURE_SET:1" value1="iTRAQ4PLEX"/>
</LIST>
```

The following colors are used below to indicate examples of the three major types of quantitation:

- **Blue** – isobaric MS/MS type.
- **Red** – MS-based quantitation, where the light state is unlabeled.
- **Green** – MS-based quantitation, where both heavy and light are labeled.

The QUANT_TYPE elements are in the next section of the Parameter Translation file, following the user input translations. For simplicity, only the sets corresponding to the three representative cases in the **Sample Type** list are shown:

```
<QUANT_TYPE xml:id="iTRAQ8PLEX" data_type="MSMS">
    <QUANT_LABEL name="IT113" display="113"/>
    <QUANT_LABEL name="IT114" display="114"/>
    <QUANT_LABEL name="IT115" display="115"/>
    <QUANT_LABEL name="IT116" display="116"/>
    <QUANT_LABEL name="IT117" display="117"/>
    <QUANT_LABEL name="IT118" display="118"/>
    <QUANT_LABEL name="IT119" display="119"/>
    <QUANT_LABEL name="IT121" display="121"/>
</QUANT_TYPE>


    <QUANT_TYPE xml:id="SILAC (Lys+6, Arg+10)" data_type="MS" separate_searches="true">
        <QUANT_LABEL name="Light" display="L" unmods="MOD_FEATURE_SET:122"/>
        <QUANT_LABEL name="Heavy" display="H" mods="MOD_FEATURE_SET:97"/>
    </QUANT_TYPE>


<QUANT_TYPE xml:id="ICAT9" data_type="MS" separate_searches="true">
    <QUANT_LABEL name="light" display="L" mods="MOD_FEATURE_SET:93"/>
    <QUANT_LABEL name="heavy" display="H" mods="MOD_FEATURE_SET:94"/>
</QUANT_TYPE>
```

The following table describes the elements and attributes of the QUANT_TYPE element in the ParameterTranslation.xml file.

Elements and Attributes for the QUANT_TYPE Element	
Element or Attribute	**Description**
QUANT_TYPE	One set per quant method.
xml:id	The link to invoke this method from a Sample Type list ITEM.

Elements and Attributes for the QUANT_TYPE Element	
Element or Attribute	**Description**
data_type	● MSMS: for isobaric quantitation derived from MS/MS spectra ● MS: for any other type of quantitation
separate_searches	If present and set to *true,* this attribute causes separate searches to be run for light and heavy label modifications. Examples where this attribute is set to true are samples labeled with the ICAT, SCIEX mTRAQ™, and SILAC reagents.
QUANT_LABEL	One per quantitation channel. There are always two QUANT_LABEL elements for MS-based quant, and either 4 or 8 for samples labeled with SCIEX iTRAQ® reagents.
name	A longer name for the label. This name is shown in the Denominator list in the Protein Quant and Summary Statistics tabs in the ProteinPilot software. For MS-based quantitation, this name also appears in the highlight bar labels in the Peptide Quantitation Information plot.
display	The short name for a label channel, mainly shown in column headers in the ProteinPilot software results.

Elements and Attributes for the QUANT_TYPE Element	
Element or Attribute	**Description**
mods, unmods	● **For MS-based duplex, where heavy and light are modified** – Use the *mod* attribute to specify the modification set for the label for each QUANT_LABEL element. The *unmod* attribute is unused. ● **For MS-based duplex, where the light form is unmodified** – Use the *mod* attribute to specify the modification set for the heavy QUANT_LABEL element. For the light QUANT_LABEL element, use the *unmod* attribute to indicate that the light state is the lack of the heavy label. For example, in the RED ms-based quantitation scheme above, there must also be a Feature set for the unmodified light form listed in the <!-- SILAC unmods - dummy mod sets to define the unlabeled states --> section of the Parameter Translation file.<MOD_FEATURE_SET xml:id="MOD_FEATURE_SET:122" name="dummy light SILAC Lys +6, Arg +10"> <MOD_FEATURE mod="Label:13C(6)"> <OCCURRENCE target="Lysine" prob="0.001"/> </MOD_FEATURE> <MOD_FEATURE mod="Label:13C(6)15N(4)"> <OCCURRENCE target="Arginine" prob="0.001"/> </MOD_FEATURE> </MOD_FEATURE_SET> ● **For MS/MS-based isobaric quantitation** – Neither *mod* nor *unmod* is used. The modification set for the label is specified by the Sample Type list's ITEM element under USER_INPUT_TRANSLATIONS.

New with the ProteinPilot 5.0 software, when creating an MS1-based quant scheme where the light form is unlabeled (for example, any kind of SILAC), a “dummy mod” set must be created for the mechanics to work correctly. Consider the following example:

This block defines the quant scheme:

```
<QUANT_TYPE xml:id="SILAC (Lys+4, Arg+10)" data_type="MS" separate_searches="true">
    <QUANT_LABEL name="Light" display="L" unmods="MOD_FEATURE_SET:121"/>
    <QUANT_LABEL name="Heavy" display="H" mods="MOD_FEATURE_SET:117"/>
</QUANT_TYPE>
```

Further down the file, there are mod sets for both the light and heavy labeling parts of the SILAC scheme. The heavy set has relatively high probabilities for the mods, while the light form has very low probabilities.

```
<MOD_FEATURE_SET xml:id="MOD_FEATURE_SET:121" name="dummy light SILAC Lys +4, Arg
+10">
    <MOD_FEATURE mod="Label:13C(4)">
        <OCCURRENCE target="Lysine" prob="0.001"/>
    </MOD_FEATURE>
    <MOD_FEATURE mod="Label:13C(6)15N(4)">
        <OCCURRENCE target="Arginine" prob="0.001"/>
    </MOD_FEATURE>
</MOD_FEATURE_SET>

<MOD_FEATURE_SET xml:id="MOD_FEATURE_SET:117" name="SILAC Lys +4, Arg +10">
    <MOD_FEATURE mod="Label:13C(4)">
        <OCCURRENCE target="Lysine" prob="0.98"/>
    </MOD_FEATURE>
    <MOD_FEATURE mod="Label:13C(6)15N(4)">
        <OCCURRENCE target="Arginine" prob="0.98"/>
    </MOD_FEATURE>
</MOD_FEATURE_SET>
```

Adding a New Digestion Set

Creating a new digestion set definition is easy. However, it is a greater challenge to determine the estimates of the probability values for an optimal set, one that can take the greatest advantage of the Paragon algorithm.

Digestion agents are described in two places. In the Data Dictionary, where the element is referred to as a cleavage agent, digestion agents are described in terms of binary rules. That is, between any two amino acid residues, does the digestion agent always cleave or never cleave? In the Parameter Translation file, digestion agents are described in terms of probabilities. That is, between any two amino acid residues, what fraction of the time does the digestion agent cleave? Fraglet uses the Data Dictionary definition, while Taglet uses the Parameter Translation file definition. In general, when a digestion set is defined, make sure to provide both definitions.

Adding a New Digestion Set: General

Add a new digestion set

1. Verify that the required cleavage agent is already defined in the Data Dictionary, or add the cleavage agent to the Data Dictionary, if necessary.
2. Create a new digestion set by creating a new DIGEST_SET element with a unique xml:id.

   NOTE: Users might want to copy a similar, existing MOD_FEATURE_SET element set, and then edit the copy.

3. Connect the user interface to the new DIGEST_SET element by editing the Digestion list.

   Refer to Connecting a Workflow Parameter Set to the User Interface on page 12.

4. Save and close all of the files, and then restart the ProteinPilot software.

Adding a New Digestion Set: Details

The following are some entries for cleavage agents in the Data Dictionary:

```
<CAgt rKey="0">
    <Nme>Asp N</Nme>
    <TLC>AN</TLC>
    <CB>Aspartic Acid</CB>
</CAgt>

<CAgt rKey="0">
    <Nme>CNBr</Nme>
    <TLC>Cb</TLC>
    <CA>Methionine</CA>
    <CMd>Met-&gt;Hsl</CMd>
    <CMd>Met-&gt;Hse</CMd>
</CAgt>

<CAgt rKey="0">
    <Nme>Chymotrypsin</Nme>
    <TLC>C</TLC>
    <CA>Isoleucine</CA>
    <CA>Leucine</CA>
    <CA>Phenylalanine</CA>
    <CA>Tryptophan</CA>
    <CA>Tyrosine</CA>
    <CNB>Proline</CNB>
</CAgt>

<CAgt rKey="0">
    <Nme>Glu C(V8 Protease)</Nme>
    <TLC>G</TLC>
    <CA>Aspartic Acid</CA>
    <CA>Glutamic Acid</CA>
    <CNB>Proline</CNB>
</CAgt>
```

The following table describes the elements and attributes for a cleavage agent in the ProteinPilot.DataDictionary.xml file.

Elements and Attributes for the CAgt Element	
Element or Attribute	**Description**
CAgt	Defines a cleavage agent in the Data Dictionary (“Cleavage agent”).
Nme	The name of the cleavage agent.
TLC	Not used.
CA	The residue after which the cleavage agent cleaves. It must match an amino acid definition in the Data Dictionary.
CB	The residue before which the cleavage agent cleaves. It must match an amino acid definition in the Data Dictionary.
CNB	A residue where the cleavage agent does not cleave (“Cleaves not before”). It must match an amino acid definition in the Data Dictionary.
HB	Not used by the ProteinPilot software. A residue where cleavage agent is hindered (“Hindered before”). It must match an amino acid definition in the Data Dictionary.
CMd	A modification that results from the cleavage (“Cleavage modification”).

The following examples are digestion set definitions from the Parameter Translation file. The first example is relatively complicated, while the second is simpler.

```
<DIGEST_SET xml:id="DIGEST_SET:2" name="Average chymotrypsin set" default_prob="0.001">
    <DIGEST_AGENT agent="Chymotrypsin" miscleavage_factor="0.75"/>
    <CLEAVAGE_RULE after="Leucine" prob="0.55">
        <EXCEPTION before="Proline" prob="0.03"/>
    </CLEAVAGE_RULE>
    <CLEAVAGE_RULE after="Isoleucine" prob="0.35">
        <EXCEPTION before="Proline" prob="0.03"/>
    </CLEAVAGE_RULE>
    <CLEAVAGE_RULE after="Phenylalanine" prob="0.85">
        <EXCEPTION before="Proline" prob="0.03"/>
    </CLEAVAGE_RULE>
    <CLEAVAGE_RULE after="Tryptophan" prob="0.90">
        <EXCEPTION before="Proline" prob="0.03"/>
    </CLEAVAGE_RULE>
    <CLEAVAGE_RULE after="Tyrosine" prob="0.95">
        <EXCEPTION before="Proline" prob="0.03"/>
    </CLEAVAGE_RULE>
```

```
    <CLEAVAGE_RULE after="Proline" prob="0.01">
        <EXCEPTION before="Proline" prob="0.03"/>
    </CLEAVAGE_RULE>
</DIGEST_SET>

<DIGEST_SET xml:id="DIGEST_SET:5" name="CNBr set" modifications="MOD_FEATURE_SET:131"
default_prob="0.001">
    <DIGEST_AGENT agent="CNBr" miscleavage_factor="0.75" />
        <CLEAVAGE_RULE after="Methionine" prob="0.80">
            <EXCEPTION before="Proline" prob="0.03"/>
        </CLEAVAGE_RULE>
    </DIGEST_SET>
```

The None set, which specifies no digestion, is the simplest of all digestion sets, with no elements.

```
<DIGEST_SET xml:id="DIGEST_SET:3" name="None" default_prob="0.01"/>
```

For space reasons, smaller digest sets are shown here. However, the definition for Trypsin in the Parameter Translation file is an example of how these sets can be very complex.

The following table describes the elements and attributes of the DIGEST_SET element in the ParameterTranslation.xml file.

Elements and Attributes for the DIGEST_SET Element

Element or Attribute	Description
DIGEST_SET	One set per digestion method. To ensure support for serial or parallel digestion workflows, the workflows must be defined in a single DIGEST_SET.
xml:id	Must be unique.
name	This name identifies the element but does not appear in the user interface. Use it as appropriate for notes or comments.
default_prob	The background probability level for all cleavages other than those explicitly specified in the set. This setting controls the baseline specificity of a digest method. This is the only probability for the “None” DIGEST_SET, the “no enzyme” setting appropriate for peptidome studies. We do not recommend setting this value higher than 0.01.
DIGEST_AGENT	Indicates the cleavage agent in the Data Dictionary. This is required for any definition to be used in a Rapid ID search. Refer to Digestion Probabilities on page 44.

Elements and Attributes for the DIGEST_SET Element	
Element or Attribute	**Description**
agent	The *Nme* element for the corresponding *CAgt* element in the Data Dictionary. This connects the DIGEST_SET to its definition in the Data Dictionary.
miscleavage_factor	A factor influencing the number of missed cleavages which the search considers. ● For the Taglet search, *miscleavage_factor* is not used. ● For the Fraglet search, the algorithm estimates p, the probability of a missed cleavage, with the formula p = (*miscleavage_factor*)^n, where "n" is the number of missed cleavages. Only n=1 and n=2 are considered. If the probability is greater than the Fraglet threshold, the search will look for peptides resulting from n missed cleavages. For example, if *miscleavage_factor* is 0.75, the probability of 2 missed cleavages is estimated as $(0.75)^2 = 0.562$. This setting does not have a strong influence on results.
CLEAVAGE_RULE	The probability of cleavage after a certain amino acid. Specify any exceptions with EXCEPTION subelements.
EXCEPTION	An exception to the CLEAVAGE_RULE. The combination of a parent CLEAVAGE_RULE and child EXCEPTION rule indicates the probability for an exact pair of amino acids.

A NOTE ABOUT DIGESTION METHODS THAT CLEAVE BEFORE A RESIDUE

Although the syntax is natural for digestion agents that primarily cleave after certain amino acids such as trypsin, the syntax is not so natural for digestion agents that primarily cleave before certain amino acids, such as Asp-N.

For digestion agents like Asp-N, create a CLEAVAGE_RULE element for every possible amino acid specifying a low probability, and each of the CLEAVAGE_RULE elements must have an EXCEPTION subelement specifying the amino acids that the digest agent cleaves before with high probability. Refer to DIGEST_SET:9 for Asp-N for an example.

Adding a New Species Set

Adding a New Species Set: General

Adding a new species set is a relatively simple parameter translation change. The steps differ slightly from the other types of parameter translation changes.

Add a new species set

1. Add a new ITEM to the Species LIST element in the USER_INPUT_TRANSLATIONS element.
2. Set the *name* attribute for the ITEM.

   The *name* attribute is what will appear in the Species list in the Paragon™ Method dialog. The ProteinPilot software convention is to use the Latin name for the species.
3. Add INTERSECTION elements to the SPECIES_SPECIES_MATRIX element.

   For each INTERSECTION element:

   - Enter a *species* attribute that is a variant of the species name which occurs in the definition line for a protein in the FASTA database.
   - For the *in* attribute, enter the value used for the *name* attribute in step 2.
   - Optionally, change the value of *prob*.
4. Save and close all of the files, and then restart the ProteinPilot software.

Adding a New Species Set – An Example

The following example shows the human species setting. The following is the list of species in the USER_INPUT_TRANSLATIONS element that defines the items in the Species list in the Paragon Method dialog:

```
<LIST name="Species">
    <ITEM name="Arabidopsis thaliana" />
    <ITEM name="Aspergillus niger" />
    <ITEM name="Bos taurus" />
    <ITEM name="Caenorhabditus elegans" />
    <ITEM name="Canis familiaris" />
    <ITEM name="Drosophila melanogaster" />
    <ITEM name="Equus caballus" />
    <ITEM name="Escherichia coli" />
    <ITEM name="Felis catus" />
    <ITEM name="Gallus gallus" />
    <ITEM name="Glycine max" />
    <ITEM name="Homo sapiens" />
    <ITEM name="Mus musculus" />
    <ITEM name="Nicotina tabacum" />
```

```
    <ITEM name="Oryctolagus cuniculus" />
    <ITEM name="Ovis aries" />
    <ITEM name="Rattus norvegicus" />
    <ITEM name="Saccharomyces cerevisiae" />
    <ITEM name="Sus scrofa" />
    <ITEM name="Xenopus laevis" />
    <ITEM name="Zea mays" />
</LIST>
```

The SPECIES_SPECIES_MATRIX element, also in the parameter file, defines how the species selection from the **Paragon Method** dialog is translated by the Paragon algorithm. Each INTERSECTION specifies the probability of finding a correct answer in a FASTA entry with the *species* attribute for the species specified by the *in* attribute. This section shows the other terms that match to Homo sapiens:

```
<SPECIES_SPECIES_MATRIX name="default" default_intraspecies="1.0" default_extraspecies="0.0" >
…
    <INTERSECTION species="Homo sapiens" in="Homo sapiens" prob="1.0" />
    <INTERSECTION species="HUMAN" in="Homo sapiens" prob="1.0" />
    <INTERSECTION species="man" in="Homo sapiens" prob="1.0" />
    <INTERSECTION species="9606" in="Homo sapiens" prob="1.0" />
…
</SPECIES_SPECIES_MATRIX>
```

This definition covers common synonyms like “man” and other notation systems for species. For example, 9606 is the Swiss-Prot *Taxon Identifier*, “HUMAN” is the Organism Identification Code or “mnemonic” name, and “homo sapiens” is the Latin name.

The software matches items in the *species* attribute using “contains” logic. If the species information in the FASTA entry contains this string, it is considered a match. This means, for example, that the Latin names of subspecies and strains also match the parent Latin name in the species set in the Parameter Translation file.

The UniProt taxonomy site at http://www.uniprot.org/taxonomy/ can be used to develop the INTERSECTION elements. For example, search for “yeast” to quickly find the information analogous to the human example, as shown in Figure 2.

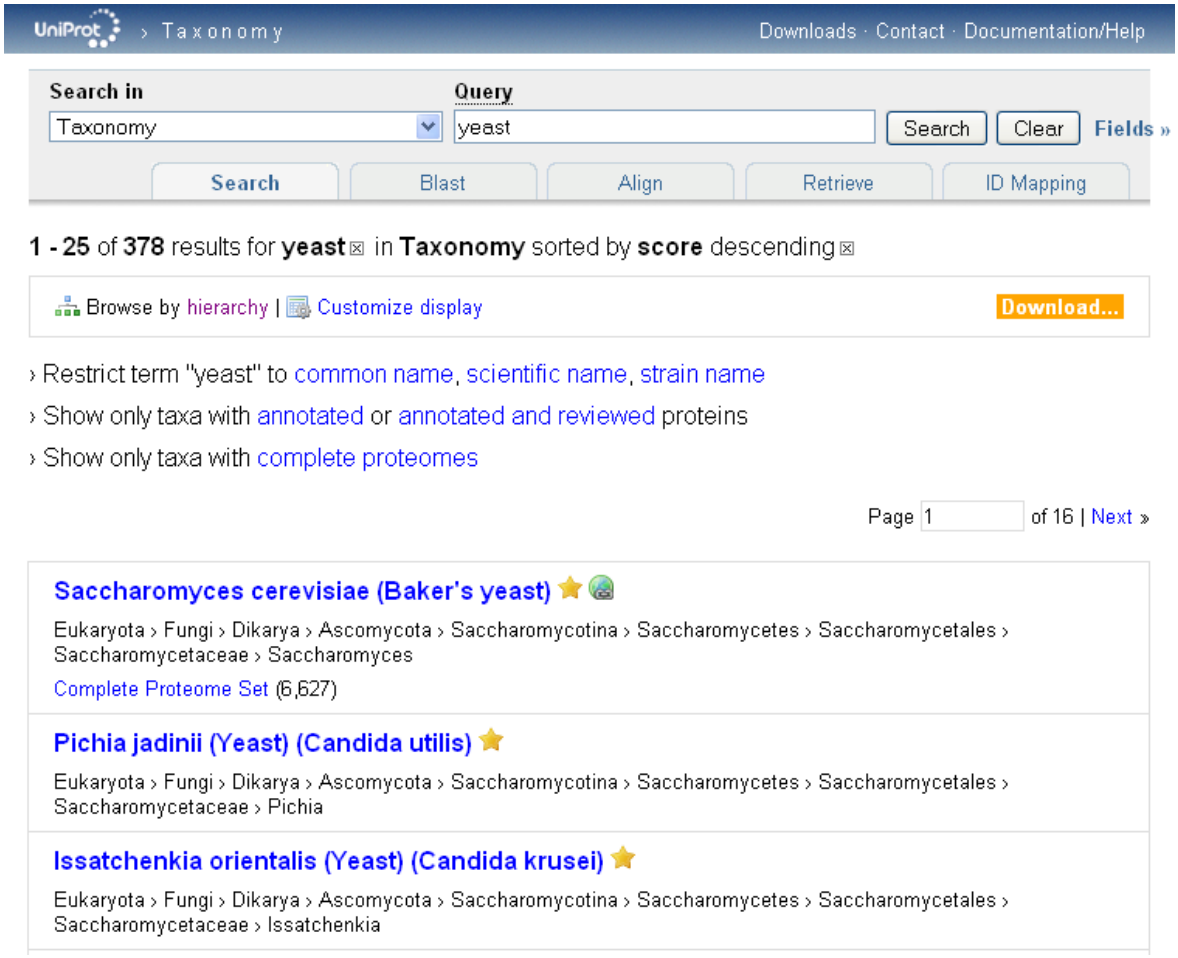

Figure 2 – Output from the UniProt website showing different names for yeast

Using Probabilities in the Species Set

In the current Parameter Translation file, the default species sets either include or exclude proteins using hard filters, since all species are set to *prob*=1.0. However, if *prob* is set to a value other than 1.0, the Taglet search considers species probabilities in the same way as other feature probabilities. This functionality can be useful when using a database that contains closely related species or subspecies, because it allows users to assign a higher probability to preferred species or subspecies.

Users can decide how complete to make the species set definition. However, to search a higher taxonomic level, for example, all fungi or all bacteria, it is often easier to select a subset of proteins at the UniProt website and export it to a

FASTA file than to manually create a complex species set in the Parameter Translation file.

Working with Instrument Definitions

Users can work with the workflow parameter sets that control how the Paragon™ algorithm treats data from different instruments. An INSTRUMENT element defines tolerances, charge states, and other details that describe the mass spectrometry data from a particular type of instrument.

The following three examples represent different categories of instruments:

```
<INSTRUMENT xml:id="INSTRUMENT:12" name="QSTAR Elite ESI" ionization="ESI"
taglet_ef="EF_CURVE:1" fraglet_correction="1" quant_threshold_sn="9">
    <AUTO_CALIBRATION ms="true" msms="true" msmstolsdratio="4" mstolsdratio="4" type="dalton"
squareroot="true" />
    <MSTOLERANCE VALUE="0.2" TYPE="dalton"/>
    <MSMSTOLERANCE VALUE="0.2" TYPE="dalton"/>
    <MS_STANDARD_DEVIATION VALUE="0.0076" TYPE="dalton"/>
    <MSMS_STANDARD_DEVIATION VALUE="0.015" TYPE="dalton"/>
    <DEFAULT_CHARGE_STATE charge="2"/>
    <DEFAULT_CHARGE_STATE charge="3"/>
    <DEFAULT_CHARGE_STATE charge="4"/>
</INSTRUMENT>

<INSTRUMENT xml:id="INSTRUMENT:21" name="4000 QTRAP ESI" ionization="ESI"
taglet_ef="EF_CURVE:2" fraglet_correction="1.5" quant_threshold_sn="6">
    <AUTO_CALIBRATION ms="true" msms="true" msmstolsdratio="3" mstolsdratio="4" type="dalton" />
    <MSTOLERANCE VALUE="0.7" TYPE="dalton"/>
    <MSMSTOLERANCE VALUE="0.6" TYPE="dalton"/>
    <MS_STANDARD_DEVIATION VALUE="0.1" TYPE="dalton"/>
    <MSMS_STANDARD_DEVIATION VALUE="0.12" TYPE="dalton"/>
    <DEFAULT_CHARGE_STATE charge="2"/>
    <DEFAULT_CHARGE_STATE charge="3"/>
    <DEFAULT_CHARGE_STATE charge="4"/>
</INSTRUMENT>

<INSTRUMENT xml:id="INSTRUMENT:1" name="4700" ionization="aMALDI" taglet_ef="EF_CURVE:1"
fraglet_correction="1" quant_threshold_sn="9">
    <AUTO_CALIBRATION ms="true" msms="true" msmstolsdratio="3" mstolsdratio="4" type="dalton"
squareroot="true" />
    <MSTOLERANCE VALUE="0.15" TYPE="dalton"/>
    <MSMSTOLERANCE VALUE="0.4" TYPE="dalton"/>
    <MS_STANDARD_DEVIATION VALUE="0.08" TYPE="dalton"/>
    <MSMS_STANDARD_DEVIATION VALUE="0.15" TYPE="dalton"/>
    <DEFAULT_CHARGE_STATE charge="1"/>
</INSTRUMENT>
```

The following table describes the elements and attributes of the INSTRUMENT element in the ParameterTranslation.xml file.

Elements and Attributes for the INSTRUMENT Element	
Element or Attribute	**Description**
INSTRUMENT	One INSTRUMENT element defines one instrument parameter set. Users can specify only one instrument per search in the Paragon method; however, the software automatically recognizes native AB SCIEX formats, enabling combined instrument searches. Because the translation between data type and instrument set is based on the xml:id, an important repercussion of this automatic instrument recognition is that, when the settings for an AB SCIEX instrument are modified, the existing INSTRUMENT set must be modified, as opposed to creating a new one.
xml:id	Must be unique.
name	Not required, but it is good practice to have a descriptive name to help readability. The text that appears in the instrument list in the Paragon™ Method dialog is defined by the *name* attribute for the ITEM element in the Instrument LIST under USER_INPUT_TRANSLATIONS.
ionization	Not used.
taglet_ef	The allowed values for the *taglet_ef* attribute are: ● EF_CURVE:1 - for medium to high resolution MS/MS data ● EF_CURVE:2 - for lower resolution ion trap MS/MS data
fraglet_correction	A multiplier applied to the Fraglet cutoff probability (0.1) to determine which modifications are searched for in a Fraglet search. Estimated guidelines for the value of *fraglet_correction* are: ● Low resolution MS: 1.5 (Fraglet searches for fewer modifications) ● Medium resolution MS: 1.0 ● High resolution MS: 0.8 (Fraglet searches for more modifications) Refer to Modification Probabilities on page 42.
quant_threshold_sn	The signal-to-noise (S/N) threshold for reporting a SCIEX iTRAQ® reagent ratio. For any ratio, if the sum of the S/N of the numerator and the S/N of the denominator falls below this threshold, the ratio is not reported (the corresponding cell in the Peptide Quantitation table is blank). The S/N is peak area divided by peak area error.

Elements and Attributes for the INSTRUMENT Element	
Element or Attribute	**Description**
AUTO_CALIBRATION	Controls whether automatic recalibration is performed. Recalibration is based on the high-scoring peptides obtained from an initial fast search. When *ms*=true, recalibration is performed for the parent ion data. When *msms*=true, recalibration is performed for the product ions. The two controls can be set independently.
type	The unit of measure for tolerances – either 'dalton' or 'ppm'. While the instrument sets provided in ProteinPilot 5.0 only use Daltons as the tolerance units, 'ppm' may also be used.
mstolratio	After recalibration, the *m/z* tolerance used by Fraglet for matching precursor ion *m/z* (observed and theoretical) is given by multiplying mstolratio and MS_STANDARD_DEVIATION VALUE. Not used if recalibration is off (*ms*=false) for the precursor data. ***Note:*** Taglet does not use strict tolerances, so the search might identify peptides with deltas much larger than the tolerance if tag evidence justifies it.
msmstolratio	After recalibration, the *m/z* tolerance for matching fragment ion *m/z* (observed and theoretical) is given by multiplying msmstolratio and MSMS_STANDARD_DEVIATION VALUE. Not used if recalibration is off (*msms*=false) for the fragment ion data.
squareroot	An optional attribute that, when set to true, causes recalibration analysis to be done plotting the delta of the square root of the observed *m/z* and square root of the theoretical *m/z* vs. the square root of the theoretical *m/z*. This calibration approach is likely to be more accurate for TOF-based instruments than the default approach (if this setting is not specified) of plotting the delta of observed *m/z* minus the theoretical *m/z* versus the theoretical *m/z*.
MSTOLERANCE VALUE	The *m/z* tolerance for matching precursor ion *m/z* (observed and theoretical) for: ● searches without MS recalibration ● the initial fast search upon which MS recalibration is based The *type* attribute suggests other calibration models can be used. However, Dalton is the only option at present.

Elements and Attributes for the INSTRUMENT Element	
Element or Attribute	**Description**
MSMSTOLERANCE VALUE	The *m/z* tolerance for matching fragment ion *m/z* (observed and theoretical) for: ● searches without MS/MS recalibration ● the initial fast search upon which MS/MS recalibration is based The *type* attribute suggests other calibration models can be used. However, Dalton is the only option at present.
MS_STANDARD_ DEVIATION VALUE	The standard deviation of delta *m/z* errors on the precursors expected for properly calibrated data for this instrument type. After recalibration, the *m/z* tolerance for matching precursor ion *m/z* (observed and theoretical) is given by multiplying mstolratio and MS_STANDARD_DEVIATION VALUE. This element is not used if recalibration is off (*ms*=false) for the precursor data. False discovery rate analysis can be used to optimize the value for MS_STANDARD_DEVIATION VALUE for the quality of the data acquired in the lab.
MSMS_STANDARD_ DEVIATION VALUE	The standard deviation of delta *m/z* errors on the fragment ions expected for properly calibrated data for this instrument type. After recalibration, the *m/z* tolerance for matching fragment ion *m/z* (observed and theoretical) is given by multiplying msmstolratio and MSMS_STANDARD_DEVIATION VALUE. This element is not used if recalibration is off (*msms*=false) for the precursor data. False discovery rate analysis can be used to optimize the value for MSMS_STANDARD_DEVIATION VALUE for the quality of the data acquired in the lab.
DEFAULT_CHARGE_ STATE	This element defines a charge state that is part of the default set of charges that are considered when there is uncertainty regarding the actual charge state. This can occur when: ● The data is low resolution ion trap data and there is no enhanced resolution scan. ● There is disagreement between the charge states determined by the instrument software and the ProteinPilot software. ● An .mgf file indicates more than one charge (currently the indications are ignored and the full range specified by the DEFAULT_CHARGE_STATE elements is searched).

Adding a New Substitution Set

The following is a small section of the original single substitution set used in the ProteinPilot software, versions 1 through 4.5 Beta.

```
<SUBSTITUTION_SET xml:id="SUBSTITUTION_SET:1" name="Crudely based on BLOSUM62">
…
    <SUBSTITUTION_FEATURE residue="A" sub="R" probability="0.0015"/>
    <SUBSTITUTION_FEATURE residue="R" sub="A" probability="0.0015"/>

    <SUBSTITUTION_FEATURE residue="A" sub="N" probability="0.001"/>
    <SUBSTITUTION_FEATURE residue="N" sub="A" probability="0.001"/>
…
</SUBSTITUTION_SET>
```

Read the second line as “Residue *A* is replaced in the same location by the residue *R*, with a probability of *0.0015* per occurrence of alanine.”

Since ProteinPilot 5.0 software, there are multiple substitution set based on different assumptions about the physical origin of substitutions. There are a few common features to these sets:

- They do not include the most unlikely substitutions. That is, there are not probability values for all cells in the matrix.
- Substitution of Ile/Leu, in either direction, is not included in the default set because they are not differentiable by most mass spectrometry experiments.
- Substitutions equivalent to the two deamidation modifications (N converting to D and Q converting to E) are not included because deamidation is more likely. The substitutions corresponding to amidation are also not included because a one Dalton shift is more likely due to incorrect peak picking.

Similar to the other elements in the Parameter Translation file, users can either edit the current SUBSTITUTION_SET or create a new one, adding an entry to the LIST for Identification focus to make the set available in the Paragon Method dialog.

Setting Feature Probabilities

The feature probability measures how often users expect to see the feature in samples. The Parameter Translation file allows users to set probabilities for modifications, amino acid substitutions, and digestion. The feature probabilities are estimates and do not need to be highly accurate in general. Estimates provide sufficient guidance to the Paragon algorithm for most modifications. However, there are some important boundary effects to be aware of that are described in this section.

ALLOWED VALUES FOR FEATURE PROBABILITY

Feature probabilities must range between 0 and 1.0, to indicate the fraction of possible occurrences where the feature actually happens. For example, if the feature is a modification on tyrosine, then the feature probability should be the fraction of all tyrosines where this modification occurs in the sample type of interest. Think of feature probabilities as estimated global averages for the type of sample of interest, not as an exact measure of what will be observed. A modification probability of 0.001 for diphthamide on histidine means that 0.001 of the histidine residues in the sample are expected to be modified with this feature. At the opposite extreme, a modification probability of 0.99 for carbamidomethyl on cysteine means that virtually all of the cysteine residues in the samples are expected to be carbamidomethylated. This makes sense only if the sample preparation workflow includes a step in which the user alkylates the cysteines with iodoacetamide.

THE MAXIMUM FEATURE PROBABILITY IS USED

If a workflow includes parameter sets with conflicting feature probabilities, then the maximum value is used.

For example, for a search with the Special Factor Gel-based ID check box selected, the standard workup modification set (MOD_FEATURE_SET:1), which is used by all workflows, assigns a probability of 0.15 to oxidation of methionine, and the gel-based ID modification set (MOD_FEATURE_SET:44) assigns a probability of 0.4 to oxidation of methionine. The search uses the 0.4 value. The general role of feature probabilities is to layer in new modifications and given higher probabilities to features already in default sets.

Modification Probabilities

Although modification probabilities should generally be set to their true prior value, it can be helpful to understand how these probabilities are used by the Paragon algorithm. The mechanics of the algorithm cause some boundary effects that users need to be aware of when dealing with any features that occur more often than 1/100.

Is the Modification Fixed?

If a modification has probability p >0.5, then the modification is considered "fixed" and is applied to every instance of the amino acid residue or chain terminus. In addition, a "pseudo-modification" is created (that is, a modification representing the absence of the fixed modification) with a probability of $1 - p$.

Making a modification fixed increases computational efficiency. The drawback is that the Paragon algorithm searches only for the fixed modification and its absence; no other modification on that residue/terminus is searched for. This is generally a problem only if users want to find both natural and artifact modifications on cysteine when cysteine alkylation is part of the workflow. If users want to search for multiple modifications on the same residue, make sure that none of the modifications has a probability greater than 0.5. For example, to look for many natural post-translational modifications on cysteine when the workflow employs cysteine alkylation, set the probability for the cysteine alkylation modification artificially low to 0.49. This might result in a small cost in search time and discrimination. Since the ProteinPilot 4.0 software, there are options that support this use directly. Refer to the *Paragon™ Method Settings Guide* in the Help folder.

Modification Probabilities are Rescaled

After the decision has been made as to whether a modification is fixed, all modification probabilities p (including the probability of absence of a fixed modification) are rescaled to $p / (1 - p)$. This rescaling has minimal impact for low-probability modifications. For high-probability modifications, this rescaling raises the maximum possible probability value from 0.5 to 1. It also provides a mechanism, when appropriate, for preventing joint probabilities from dropping rapidly as the number of modifications increases, when considering more than one modification on a peptide hypothesis. The following examples illustrate this:

- In the standard Biological Modifications set, the modification probability for glutathione on cysteine is 0.033. The rescaled probability is 0.033 / (1 – 0.033) = 0.034126, which is close to the original probability value of 0.033.
- In the standard MMTS set, the modification probability for methylthio on cysteine is 0.925, which means it is a fixed modification. The probability for the absence of methylthio on cysteine is 1 – 0.925 = 0.075, and the rescaled probability of absence is 0.075 / (1 – 0.075) = 0.081081.
- In the Gel-based ID modification set, the modification probability for oxidation on methionine is 0.4. The rescaled probability is 0.4 / (1 – 0.4) = 0.666666. If a peptide hypothesis contains multiple methionine oxidations, then the individual methionine modification probabilities are multiplied together, and the relatively high rescaled probability value of 0.666666 prevents the overall probability from dropping rapidly as the number of methionine oxidations increases.

DOES FRAGLET LOOK FOR THE MODIFICATION?

The Fraglet and Taglet searches are complementary. In general, Taglet is more effective at identifying high quality spectra, and Fraglet is more effective at identifying low quality spectra. To study a specific modification expected to be observed with high frequency, make sure that Fraglet looks for that modification.

NOTE: Fraglet, similar to other conventional database search engines, works best if only common modifications are being searched. Having Fraglet search for too many rare modifications causes degradation in performance in terms of both time and result quality.

- Fraglet searches for a modification or the absence of a fixed modification only if the rescaled probability exceeds 0.1 * *fraglet_correction*. The *fraglet_correction* acts to fine tune the behavior, based on instrument type. Fraglet searches for fewer modifications for low-resolution instruments than for high-resolution instruments. Typical values for *fraglet_correction* range from 0.8 (for high-resolution instruments) to 1.5 (for low-resolution instruments).
- If a peptide hypothesis contains multiple modifications, then the modifications are assumed to be statistically independent events, and an overall probability is computed as the product of the individual rescaled modification probabilities. Fraglet scores the peptide hypothesis only if the overall probability exceeds
  0.1 * *fraglet_correction.*

Note: Thresholds are not an issue for Taglet because Taglet searches for all modifications that have non-zero probability.

DIGESTION PROBABILITIES

The digestion probabilities are used by Taglet but are ignored by Fraglet. Fraglet uses the binary digestion agent rules from the Data Dictionary.

Testing Changes to Feature Probabilities

It is strongly recommended that any changes to feature probabilities are tested.

- Does the expected feature appear at all in the results? Go to the **Spectra** tab and then sort by the **Modification** column to check.
- Does it appear in **Rapid** mode if expected?
- Does it appear with normal mass deltas, always with large deltas, or in combination with an unlikely modification?

  NOTE*:* This type of verification testing can be difficult for very low frequency events, such as rare post-translational modifications.

In addition to reviewing the results of a single search, the comparison template can be used to compare the results across multiple searches.

Defining Biological Features for the Features Tab

The **Features** tab focuses exclusively on biological features. Chemical modifications (for example, cysteine alkylations and quantitation labels) and artifact modifications (for example, results of sample workup or instrumental analysis) are not reported, although they are included where relevant as counter-evidence. The *mod_class* attribute is used to specify modifications as biological features and therefore, if detected, is listed on the **Features** tab.

To consider a modification as a biological feature and explicitly be called out as a feature on the Features tab, add the *mod_class*="BIOMOD" attribute to a MOD_FEATURE element.

```
<MOD_FEATURE MOD="OXIDATION" >

 <OCCURRENCE TARGET="CYSTEINE" PROB="0.001" MOD_CLASS="BIOMOD" />
```

Suggestions for Testing Parameter Translation Changes Using FDR Analysis

We recommend that parameter translation changes are tested with the FDR analysis function in the ProteinPilot software. The FDR analysis is essentially independent of the statistics reported by ProteinPilot software, so it provides an independent way to determine the quality of the identification results and is particularly powerful for comparison and optimization.

The **FDR Comparison_V5.0p.xlsx** file can be used to help quantitatively and visually compare multiple results. This template is located in the ProteinPilot software Help folder, typically located in:

```
C:\Program Files\AB SCIEX\ProteinPilot\Help
```

Figure 3 illustrates how the comparison template is used. FDR analysis is performed and multiple ProteinPilot Reports are generated. The summary column from each report is copied manually over to one of the input columns in the comparison template.

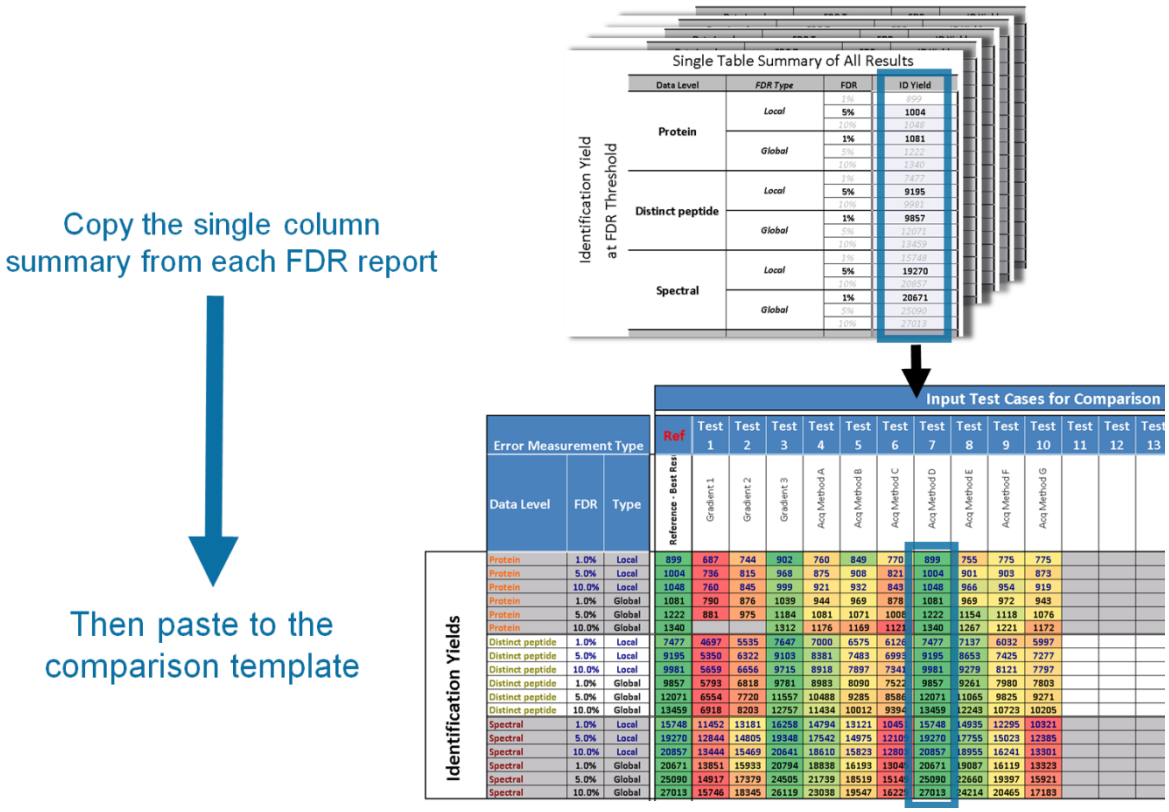

Figure 3 – Schematic illustration of the comparison template

After the comparison input is entered, the subsequent tabs in the comparison template give visual and quantitative assessment of similarity and relative performance of each input search.

Refer to the instructions on the first tab of the template file and to *False Discovery Rate Analysis with the ProteinPilot™ Software* in the Help folder.

How to Share Improvements with Others

To use the Paragon™ algorithm effectively, parameter translation is continually being improved in the ProteinPilot software. Users are encouraged to share optimizations and extensions.

There are several ways to share improvements.

- **Sharing entire files**: Give other users a copy of your ParameterTranslation.xml and ProteinPilot.DataDictionary.xml files.

  For these files to work automatically in a different installation of ProteinPilot software, the same version of the ProteinPilot software must be used. Remind users to save a copy of their own files if they have made any alterations. This approach is the safest because it ensures that the files are valid and functioning, assuming they worked for the original user. Files can still be shared if the same version of the ProteinPilot software is not used, however, the second user will need to incorporate changes into their analogous files manually.

- **Sharing only the key segments of the files**: Give other users the key sections of the files that were changed.

  This comes with the risk that some portion of the necessary changes will be missed, as well as the additional burden that subsequent users must incorporate changes into their own files. The advantage of this approach is that manually merging changes in order to maintain some existing content is easier to do if pieces are kept separate. This type of merging requires careful checking to make sure that the xml:id identifiers are unique and the XML file is valid.

- **Sharing by publication**: If you publish a paper that uses changes you have made, include the complete files as well as a file indicating the differences. Publications generally require that any customizations made to commercial software are published, and including these files is an easy way to do so.

- **Sharing by submission to the ProteinPilot Software development team** To reach the largest number of users, submit changes to the ProteinPilot software development team. Submit both intact files and a file showing only the differences. If this is not possible, make sure to indicate the differences clearly.

Inclusion of user changes in a subsequent version of the ProteinPilot software depends on whether the changes are broadly applicable and well validated. Representative data might be requested from users to support the testing process. All contributions are appreciated and, with user permission, will be acknowledged in the Parameter Translation file. Send contributions and suggestions to support@absciex.com.

Revision History

Revision	Description of Change	Date
A	First release	March 2009
B	Content updates related to ProteinPilot software v. 4.0	September 2010
C	Content updates related to ProteinPilot software v. 5.0	September 2014