Principles of Kim Plate Processing

In order to be able to use all potential of Kim one has to understand principles. The core of Kim is how it processes a plate. From that information you start understanding how to set a method definition.

The center point of Kim is a plate. It is a collection of measured data, samples and a set of defining mathematical expressions. A layout of samples (geometrical distribution of samples on a plate) together with mathematical expressions are considered a method. When you start a new plate, you assign it a method. Then you do a measurement and the plate gets a matrix of measured data. Automatic plate processing starts once data is transferred from a reader to a PC.

Plate processing consists of following steps:

- **Blanking** ... plate data normalization
- **Calibration** ... calibration curve is calculation
- **Calculations** ... common (pre-)calculations that do not depend on samples
- **Evaluations** ... calculations that are repeated for every sample on a plate
- **Validations** ... calculations that confirm whether a test was or was not valid

All steps are optional. It means that they will be carried only if they were included in a method definition of a particular plate.

In the following text we will give examples of mathematical expressions without going into details of how mathematics works in Kim. Details of mathematical engine will be dealt with in following chapters.

Blanking

This step is optional. You may choose not to do a blanking at all. System then works with raw measured data (absorbances) only. Blanking is usually not used with “quantitative” methods – methods that use calibration curve – as blanking does not contribute to calculation of sample concentrations.

Simple Blanking

This is the most usual blanking procedure. A common value is subtracted from each well on a plate. Most often you simply insert a **Blk** control in the plate layout. When you do that, the system automatically generates a blanking expression as:

```
avg( Blk )
```

The above expression tells the system to take absorbances of all **Blk** controls and to calculate their average value. The average value is then subtracted from each well of a plate. (The average value is also stored in a global variable **gblank**).

The blanking expression is not limited to calculating average of **Blk** controls. You are free to enter your own expression for a blanking value.

Once blanking is used, the system differentiates between raw data and normalized data. Each well contains two values – raw measured value (an absorbance supplied by the reader) and a normalized value – a value after blanking. With simple blanking, the normalized value is calculated by subtracting the blank from the raw value.

General Transformation

This is a more general method of data normalization. It consists of two steps. First, a common value is calculated and is assigned to a **gblank** variable (but that value is not automatically subtracted from all wells as is the case of simple blanking). Then a transforming expression is applied to each well on a plate. In the transforming expression one has to be able to refer to the original (raw) value in a well. You use a **Cur** variable to refer to that original well value. Consider following transformation:

```
Cur - gblank
```

The **Cur** variable will have different contents for each well. Actually, the above expression will have the same effect as what you get with simple blanking – a (common) blank value is subtracted from an absorbance of a well.

Using a different transformation, you can derive optical transmission from absorbances:

```
pow10( -Cur )
```

10 powered by a reverse absorbance gives an optical transmission – a ratio between light intensity after passing a well to original light intensity.

Calibration

Calibration is also an optional step. A calibration curve will be calculated if standards appear in the plate layout. System puts together absorbances of standards, their concentrations and a calibration model found in the calibration definition.

Since Kim version 5.12 it is also possible to use a calibration that was calculated on a different plate. Then the plate must not contain standards. See the appendix for details on “Stored calibration”.

Calculations

You may define a set of expressions that will be calculated prior to sample evaluations. A typical example is a calculation of threshold values. Those values depend on controls and are valid for a plate as a whole. It is therefore reasonable to calculate them once and refer to them in sample evaluations. The definition of calculations consists of a list of expressions. For each expression we also introduce a name of a variable. System calculates expressions and assigns calculated values into respective variables. In the following example we calculate two thresholds that will separate negative, equivocal and positive samples. First a middle point (Middle) between positive and negative controls is calculated. Then the lower threshold (Thr1) is calculated as being 5% less and the higher threshold (Thr2) as being 10% above the middle point:

```
Middle = ( avg(Pos) + avg(Neg) ) / 2

Thr1 = 0.95 * Middle

Thr2 = 1.1 * Middle
```

Evaluations

Evaluations are formally similar to calculations – they are defined as a list of expressions and names of variables. But they are calculated repeatedly for each (patient) sample found on a plate. Thus each sample gets a list of its own variables. (Each sample has the same list of variables but those variables have different values). There is yet another significant difference to general calculations: Evaluations are calculated “in context” of a specific sample. This means (for example) that a variable of **od**, which contains absorbance(s) of a sample, is different for each sample.

The last variable in evaluations list is considered a result of sample evaluation. It is the value that is presented in the result list.

In the following example we assume that thresholds Thr1 and Thr2 were calculated in general calculations:

```
result = category( "neg", Thr1, "+/-", Thr2, "pos", od, "?" )
```

Category is an example of a Kim built-in function. It takes the one before last numerical argument (**od** in this case) and compares it to thresholds found in even positions (Thr1, Thr2 in this case). Processing goes from left to right. When a threshold is higher then the compared value, processing stops and the function gives back (returns) a text it founds before that thresholds. The text is then assigned to a variable (**result** in this case). If the compared value is higher then all thresholds, the text after the last threshold is returned.

Assume thresholds **Thr1**=0.5 and **Thr2**=0.6.

Let one particular sample has its **od** = 0.3. Then that sample will get **result** = "neg".

Validations

Validations are formally similar to general calculations. You define a list of expressions and names of variables. Validations are calculated as the last step of a test processing. Each expression may evaluate either to a logical TRUE (in Kim, any number which is not 0 is interpreted as TRUE) or to a logical FALSE (in Kim, 0 is interpreted as FALSE).

Kim considers a test valid if all validations were TRUE. A text “OK” in the Kim status line marks this situation.

If any validation expression gives FALSE, then the test is considered invalid and a text “NOT” in red color is shown in the status line.

The flag of valid / invalid test is also given in all printouts.

Validation is also an optional step in a plate processing. If you specify no validation expression, Kim calculates a test as usual but gives no flag for test validity.