# Scope

***[Define the scope of the requirements]***

Software name: DASCOMBAT

Software development plan: 210227RW20010 Software Development Plan DASCOMBAT

Software safety class: **C**

# Introduction

This document defines the software requirements for the R-package DASCOMBAT. This R-package is used for batch correction, typically as part of a DASNORMALIZE module. The input of the R Package is a pre-processedkinase activity profile measured on DAS and REF using any PAMCHIP. The input is received via the R PACKAGE ENVIRONMENT. The output of DASCOMBAT is a batch corrected version of the input data.

The schematic below provides an overview of the DASCOMBAT R-package architecture.

**Input**

* Data matrix A
* Corresponding batch indicator variable

**DASCOMBAT fit method**

**Output (intermediate)**

* List containing batch correction model

**DASCOMBAT applyModel method**

**Input**

* Data matrix B
* Corresponding batch indicator variable

**Output**

* Data matrix B, after applying the batch correction

DASCOMBAT exports a “fit” function that calculates a batch correction model based on a data matrix A and a corresponding batch indicator variable. It also exports an “applyModel” function that takes this batch correction model, a data matrix B with corresponding batch indicator variable and outputs a batch corrected version of this data matrix B.

Data matrix B can be identical to data matrix A but this design also allows for the batch correction model to be applied to a different data matrix than that used for fitting the model, as long as the levels of the batch indicator variable overlap. Note: this latter feature is used in the DASNORMALIZE modules to fit a model on the REF samples and apply the model to the DAS samples.

# Software requirements Analysis

The ComBat methodology is described in detail in Appendix A of “210228RW19030 Development and verification of the use of reference samples in the DAS lab”, which is refered to in some requirements as “methodology doc”.

## Software requirements [Class A, B, C]

|  |  |
| --- | --- |
| **Req.#** | **Description** |
| 1.1 | The R-package DASCOMBAT accepts a matrix that represents the phosphorylation signal, where the rows represent peptides (variables) and the columns observations. |
| 1.2 | The results for DASCOMBAT is a batch corrected version of the input data matrix (Req 002), i.e. a matrix of the same dimensions containing COMBAT corrected values. |
| 1.3 | Exceptions raised by the DASCOMBAT will be transmitted to the DAS-PLATFORM via the R PACKAGE ENVIRONMENT. |

***[Define and document software system requirements from the system level requirement]***

## Software requirements content [Class A, B, C]

### Functional and capability requirements

|  |  |
| --- | --- |
| **Req.#** | **Description** |
| 2.1.01 | SP implements a fit function that calculates a batch correction model, i.e. fits a model |
| 2.1.02 | SP implements an applyModel function thatcalculates a batch correction i.e. applies a model that was calculated by the fit function. |
| 2.1.03 | The corrected output matrix has the same format as the input data matrix (see req 1.1) |
| 2.1.10 | The batch model is defined for: No reference, mean.only = FALSE  data is the input data (see req 1.1)  fit(data, mean.only=FALSE, REF=NULL, batchFactor )  This function implements ComBat by performing two calculations   * Standardization of data (see line 32 to 50 of methodology doc) * Fitting of data for a location/scale model (see line 51 to 121 of methodology doc)   *A model is returned* |
| 2.1.11 | The batch model is defined for: No reference, mean.only = TRUE  data is the input data (see req 02)  fit(data, mean.only=TRUE, REF=NULL, batchFactor)  This function implements ComBat by performing two calculations   * Standardization of data (see line 32 to 50 of methodology doc) * Fitting of data for a location only (mean only) model, see line 151-157 of methodology doc.   *A model is returned* |
| 2.1.12 | The batch model is defined for: ref.batch, mean.only = FALSE  data is the input data (see req 1.1)  fit(data, mean.only=FALSE, REF= batchValue, batchFactor)  This function implements ComBat by performing two calculations   * Standardization of data (see line 32 to 50 of methodology doc) * Fitting of data for a location/scale model with designated reference batch “batchValue” (see line 51-121 and 144-150 of the methodology doc).   *A model is returned* |
| 2.1.13 | The batch model is defined for: ref.batch, mean.only = TRUE  data is the input data (see req 021A1)  fit(data, mean.only=TRUE, REF= batchValue, batchFactor)  This function implements ComBat by performing two calculations   * Standardization of data (see line 32 to 50 of methodology doc) * Fitting of data for a location only (mean only) model with designated reference batch “batchValue”(see line 144t o 157of methodology doc)   *A model is returned* |
| 2.1.14 | The batch model is applied to data from batches included in the fit  apply(data, MODEL= aModel, batchFactor)  Apply correction (see line 122 to 139 of methodology doc)  *A matrix is returned* |

**[Describe the functional and capability requirements that the system need to fulfill]**

* ***Performance (e.g. purpose of software, timing requirements)***
* ***Physical characteristics (e.g. code language, platform, operating system),***
* ***Computing environment (e.g. hardware, memory size, processing unit, time zone, network infrastructure) under which the software needs to perform***
* ***Need for compatibility with upgrades or multiple SOUP or other device versions***

### Software system inputs and outputs

|  |  |
| --- | --- |
| **Req.#** | **Description** |
| 2.2.1 | Input and output is retrieved using the API function of the R PACKAGE ENVIRONMENT |
| 2.2.2 | The fit function accepts:   * data: a data matrix (see req 1.1) * a factor (R variable of class factor) with length equal to the number of columns of data containing the batch indicator values. * Mean.only: logical parameter indicating if a mean only must be calculated, or a L/S model otherwise. * REF: character parameter containing a REF batch or NULL if no ref batch is used. |
| 2.2.3 | Complete arrays only. The DAS-COMBAT checks if there are any missing values. If that is the case the following exception is raised: “missing values are not allowed”. |

***[List the inputs and outputs of the system software e.g.:]***

* ***Data characteristics (e.g. numerical, alpha numeric, format)***
* ***Ranges***
* ***Limits***
* ***Defauts***

### Interfaces between the software systems and other systems

|  |  |
| --- | --- |
| **Req.#** | **Description** |
|  |  |

***[List each system interface and identify the functionality of the software to accomplish the system requirement and the interface description to match the system]***

### Software driven alarms, warnings and operator messages

|  |  |
| --- | --- |
| **Req.#** | **Description** |
|  |  |

***[Describe all requirements and functionalities that alarm, warning and operator need to fulfill within the system]***

### Security requirements

|  |  |
| --- | --- |
| **Req.#** | **Description** |
|  |  |

***[Write down the security requirements of the software system e,g,:]***

* ***Related to the compromise of sensitive information***
* ***Authentication***
* ***Authorisation***
* ***Audit trail***
* ***Communication integrity***
* ***System security/malware protection***

### User interface requirements implemented by the software

|  |  |
| --- | --- |
| **Req.#** | **Description** |
|  |  |

***[List Interface Requirements that need to be implemented by the software, see examples below***:

* ***Support for manual operations***
* ***Human equipment interactions***
* ***Constraints on personnel***
* ***Area needing concentrated human attention]***

### Data definition and database requirements

|  |  |
| --- | --- |
| **Req.#** | **Description** |
|  |  |

***[Describe data definition and database requirements below:]***

* ***Form***
* ***Fit***
* ***Functions***

### Installation and acceptance requirements of the delivered medical device software at the operation and maintenance site

|  |  |
| --- | --- |
| **Req.#** | **Description** |
|  |  |

***[Specify any installation and acceptance requirements of the delivered software at the operation and maintenance site]***

### Requirements related to methods of operation and maintenance

|  |  |
| --- | --- |
| **Req.#** | **Description** |
|  |  |

***[Identify any applicable requirements related method of operation and maintenance]***

### Requirements related to IT-network aspects

|  |  |
| --- | --- |
| **Req.#** | **Description** |
|  |  |

***[- Network alarms, warnings and operator messages***

* ***Network protocols***
* ***Handling of unavailability of network services]***

### User Maintenance Requirements

|  |  |
| --- | --- |
| **Req.#** | **Description** |
|  |  |

***[Identify means for user maintenance requirements]***

### Regulatory Requirements

|  |  |
| --- | --- |
| **Req.#** | **Description** |
|  |  |

***[Identify regulatory requirements for the software specifications]***

## Risk Control Measures in Software Requirements [Class B, C]

|  |  |
| --- | --- |
| **Req.#** | **Description** |
|  |  |

***[Identify how risk control measures should be implemented.***

***Note: These requirements might not be available at the beginning of the software development and can change as the software is designed and risk control measures are further defined.]***

# DEFiNITIONS AND ABBREVIATIONS

# APPenDIX

# SIGNATURES AND APPROVAL

***[Approval for the software requirements]***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Role** | **Function title** | **Name** | **Signature** | **Release date** |
| Author |  |  |  |  |
| Reviewer |  |  |  |  |
| Authorizer |  |  |  |  |
| QARA |  |  |  |  |

# HISTORY

***[Version history for the software requirements]***

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Remarks** | **Document owner** |
| 1.0 | 31-Dec-1999 | Initial version |  |
|  |  |  |  |

# DOCUMENT APPROVAL

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Role** | **Function title** | **Name** | **Signature** | **Release date** |
| Author | Head of DAS | Rik de Wijn |  |  |
| Authorizer | Head of DAS | Rik de Wijn |  |  |
| QARA | QARA Manager | Anja Wiersma |  |  |

# DOCUMENT HISTORY

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Remarks** | **Document owner** |
| 1.0 | 30-Sep-2018 | Initial version | Faris Naji |
| 1.1 | 4-Dec-2024 | Updated lay-out  Removed safety class column per requirement | Rik de Wijn |