We developed and implemented an infrastructure to create and maintain chemometrical classifiers for identification of more than 1000 substances. The information associated with the classifier module and its classifications (substances, batches, samples) are managed in a database. The underlying process for module definition and module creation results in a complex workflow. This workflow has been implemented as platform-independent software infrastructure. It is currently running on a Linux-based server system, realising an overall processing time of about 72h for a classifier module consisting of multiple large chemometrical models.

KEYWORDS: NIR, chemometrical classifier, validation, infrastructure, automation

INTRODUCTION

As a general rule, NIR spectroscopy requires reliable chemometrical classifiers or predictors. Classifiers are used for qualitative applications, predictors for quantitative applications. The reliability of the implemented (e.g. statistical) method must be proven by validation against the respective target application. For many applications this validation procedure must be well documented and exactly reproducible. Especially for qualitative analyses the resulting disproportionate effort to create and maintain validated and well documented chemometrical classifiers often limits the use of the method. This is particularly true for classifiers basing on several hundreds of thousands of NIR spectra which have to be updated and extended regularly.

We have developed an automated infrastructure to create and maintain classifier modules which can be used to identify a wide range of pharmaceutical substances. These modules contain multiple chemometrical models, basing on different types of spectral data. This includes 310 000 NIR spectra of solids, measured in diffuse reflection, and 90 000 NIR transflectance spectra of semisolids and fluids. This large collection of spectral data is used for calibration and validation of the particular chemometrical models, and the complete classifier module respectively. The results of the very time-consuming validation procedure are summarised in an automatically generated, revision safe PDF document which typically comprises up to several thousand pages. The accuracy and integrity of this documentation is essential for pharmaceutical applications.

Problem

Creating and maintaining valid chemometrical classifiers is a time consuming and often underestimated process which potentially limits a wider use of NIR spectroscopy. Especially qualitative classification applications require the management of spectral and administrative information regarding hundreds or thousands of samples, batches and substances.

To realise continuous, reliable and professional maintenance of a chemometrical application it is necessary to define a precise workflow. The following aspects have to be considered:

• Information management: supplier, batch, certificates, reports...
• Data acquisition: sample spectra, reference spectra...
• Data processing: quality assurance, spectra correction
• Chemometrical evaluation: order reduction, outlier analysis...
• Classifier development: implementation, compilation...
• Validation: proof of method's validity
• Test: software/compatibility test, final check...
• Documentation: development logs, validation documents
For a current classifier for pharmaceutical application this results in the following workflow (Figure 1):

The result is a rather complex process, containing several very different and often parallel tasks which vary from hours to months in processing time.

All tasks must be reliably tracked, finished and well documented. The resulting software module (classifier) must be reproducible and its development must be fully traceable at any time in the future.

**Approach**

Since commercially available chemometrical software primarily addresses to developers and to a lesser extent to production environments, we decided to implement a custom software infrastructure from scratch. Core components of this infrastructure are small but flexible command line tools which can be combined arbitrarily. These tools realise well defined data interfaces for input and output and are self-contained and thereby easy to maintain. Using scripting languages like shell scripts very complex procedures can be implemented and easily extended with this approach. The following section describes an extensive sample application basing on this approach. The realised infrastructure has been designed to create, evaluate, validate and document chemometrical classifiers for pharmaceutical applications, which are updated and released regularly. The implemented infrastructure consists of the following components:

Chemometrical Information Database (CID) is a MySQL-based issue and information management system. It comprises two main modules (Figure 2):
Data input is possible concurrently for multiple users and classifier modules. Plausibility of user input is checked against a predefined set of rules. An arbitrary version of this data structures can be combined and define a versioned set of text or XML based parameter files. These parameter files are designed to control the Chemometrical Toolchain.

Chemometrical Toolchain (Figure 3) is a set of platform independent command line programs which are optimised for multi-core computer hardware systems:

- Issue tracking: tasks, problems, improvements...
- Information management: substance, sample level...
- Change management: who, when, what...
- Traceability: test reports, analysis certificates...
- Model definition: calibration, algorithms, parameters...
- Validation definition: spectra, samples...
- Change management: who, when, what...
- Release management: parameter sets, changelogs...

Data input is possible concurrently for multiple users and classifier modules. Plausibility of user input is checked against a predefined set of rules. An arbitrary version of this data structures can be combined and define a versioned set of text or XML based parameter files. These parameter files are designed to control the Chemometrical Toolchain.

Figure 2. Two modules of the chemometrical information database (CID)

Figure 3. Chemometrical toolchain for creating a documented and validated chemometrical classifier
• correctiontool: quality assurance and correction of spectra
  • bad spectrum detection (e.g. methodological errors)
  • correction of probe influence and device influence
• pcagenerator: control of parameter file and model creation
  • additional plausibility checks of substance structure and model structure information
  • generation of parameter files
• pcatool: principal component analysis
  • actor analysis (calculation of loadings, scores...)
  • self-validation
• pcaanalyzer: model evaluation
  • calculation of statistical properties (distances, covariances...)
  • (pre-) cross-validation of models
• pcamanager: classifier module generation
  • encryption of model files, access management
  • software module compilation (classification software, classification models)
• validationtool: classifier software module validation
  • validation procedure at user API level (software, chemometrical models in combination)
• reportingtool: creation of validation reports and documentation
  • detailed PDF documents per substance class

This infrastructure is currently running on an eight-core Linux server, equipped with 32GB of memory and 8TB storage space. The typical release cycle for updated classifier modules is one to two months. Computation time for one complete run (without spectrum acquisition and model definition) is about 45 hours. The result is a software module (Microsoft Windows installer), which contains complete validation documentation and can be shipped as is.

CONCLUSION AND OUTLOOK
We developed and implemented an infrastructure which allows us to create and maintain chemometrical classifiers using an inline process. All production steps are well documented and can be reproduced accurately or with well-defined variations. The underlying toolchain allows efficient adaptions for new algorithms or even new fields of application like quantification.