How to deploy the Sample Submission Shiny App

Prerequisites

Required R version

```
R version 3.5.1
```

Required packages:

Following packages are required to run the application:

readr, pdftools, png, rmarkdown, seqinr, shiny, shinyjs, shinyBS, stringr, yaml, V8, tiff, jpeg, mailR

Required Files

Some files are needed in the App directory to run properly

Config.yaml

Cofing.yaml contains all important parameters to control the App. The typical config file looks like:

```
orders_directory: ./outputs
maximum_upload_megabytes: 40.0
user_inputs_file_name: user_inputs.yaml
available_groups:
  IMB_EW: Eva Wolf
  IMB_BU: Falk Butter
  EXT_CF: External
db_summary_file: Available_species_DBs.csv
available_proteases:
- undefined
- AspN
- GluC
- LysC
- Chymotrypsin
```

Fields description:

orders_directory: specify the output directory for all submitted forms

user_inputs_file_name: specify the file name, which will be created for each submission. It contains all form inputs and additional information from the filled form.

db_summary_file: a file containing a summary of all selectable Fasta-Databases. Such a file can be created with the function createSummaryFileOfFastaFiles().

Available_species_DBs.csv

This file contains a summary of all selectable Fasta-Databases. Such a file can be created with the function createSummaryFileOfFastaFiles(). The fasta file paths must be reachable from the shiny app. See the function documentation for more details.

Common maintenance activities

This chapter describes how to maintain and configure the application.

App and Form appearance

The most appearance aspects are defined with the CSS files.

www/style.css defines the App appearance

page_style.css defines the output Form appearance respectively

(De)activate sending an email to a customer

The App offers the ability to send an email with the filled Form to the customer. It can be activated with the R option “samplesubmission.send_email":

```
options(samplesubmission.send_email=TRUE)
```

Forwarding error messages to the client

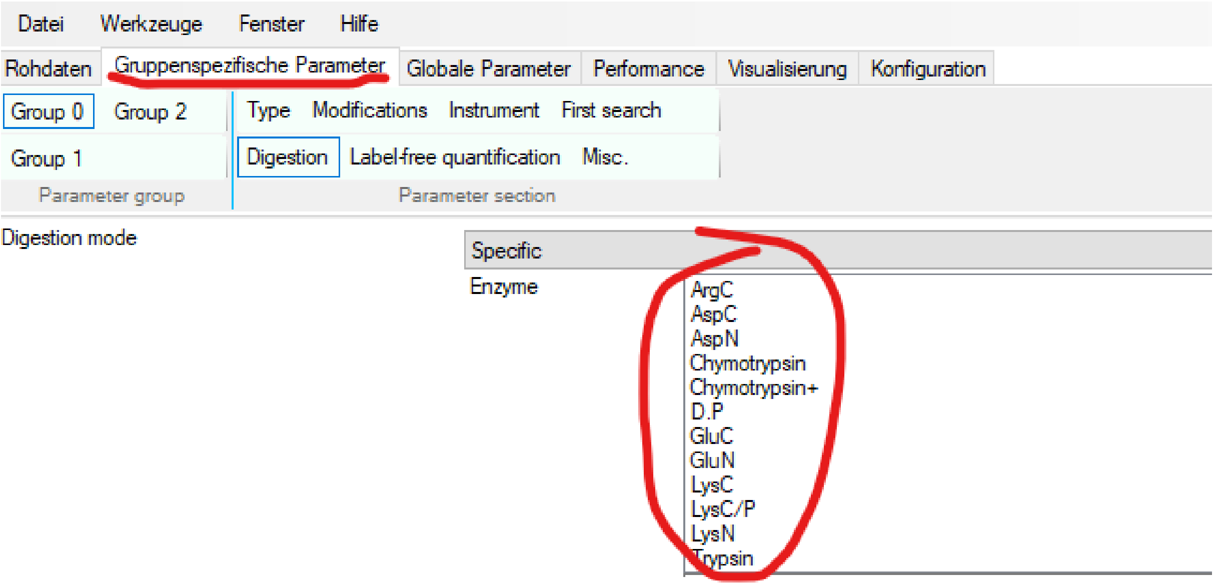
Some errors occurring during the form completing and submitting can be forwarded to the client. It can be used for debugging or user notification.

It can be activated with the R option “samplesubmission_errors_to_client":

```
options(samplesubmission.errors_to_client=TRUE)
```

Add new selectable enzymes

Besides trypsin the app offers additional enzyme selection. The available enzymes can be configured in the configuration file (see Config.yaml). To ensure a smooth automatic processing (e.g the creation of mqpar file for MaxQuant) the enzyme names should be the same as in MaxQuant software. So make sure the new enzymes names are supported by MaxQuant by looking up them inside the MaxQuant Interface:



Digestion mode: Specific

Enzyme:
ArgC
AspC
AspN
Chymotrypsin
Chymotrypsin+
D.P
GluC
GluN
LysC
LysC/P
LysN
Trypsin

Regex patterns for Protein IDs extraction

Depending on selected databases the User can choose from a list of protein IDs (Proteins the Customer looking for). These IDs are extracted dynamically from the respective fasta files. The pattern for protein ID extraction is determined with the function “getProtIdRegexFromFileName” by means of fasta file name. The function is also used within the automation script for mqpar file creation. Therefore, to extend or adjust the pattern determination function edit is needed.

Automatic Preparation for MaxQuant

This chapter describes the Usage of the script for summarizing the submission resources whit the corresponding raw files and preparation of these for MaxQuant processing.

Resources

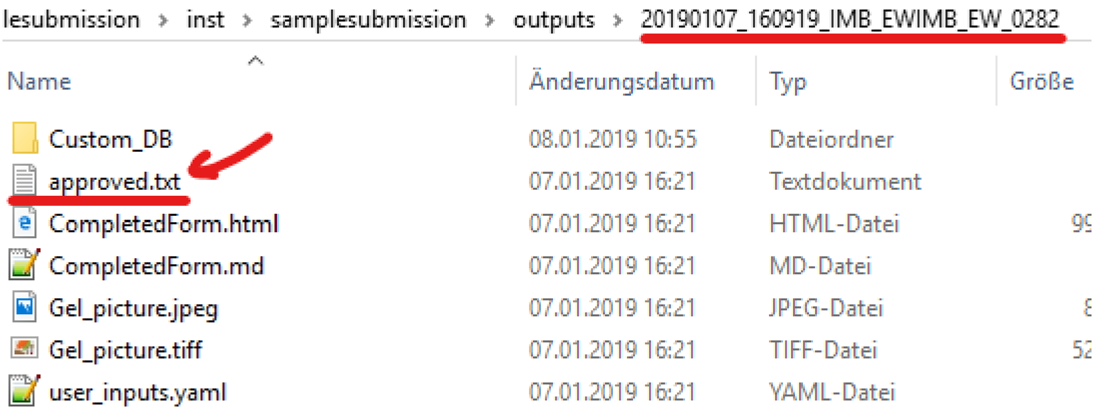
All needed resources are placed within the package directory “inst/samplesubmission”

Automation script: gatherAndPrepareForMQ.R

Maxquant parameter file template: mqpar.tmp.xml

Submission approvement

To enable the script for summarizing, each physically received submission must be receipt by creating an approvement file (approved.txt) within the submission directory.



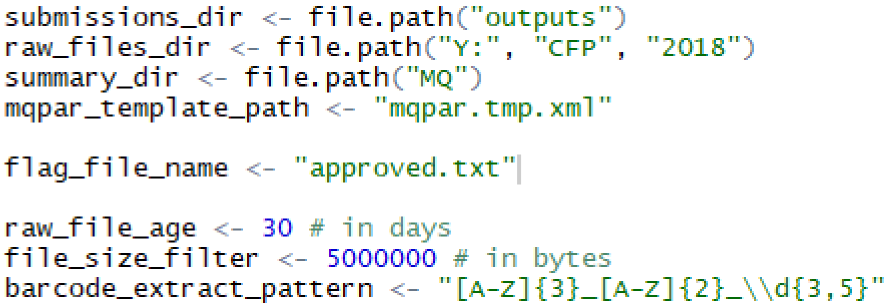
lesubmission › inst › samplesubmission › outputs › 20190107_160919_IMB_EWIMB_EW_0282

Name	Änderungsdatum	Typ	Größe
Custom_DB	08.01.2019 10:55	Dateiordner	
approved.txt	07.01.2019 16:21	Textdokument	
CompletedForm.html	07.01.2019 16:21	HTML-Datei	9¢
CompletedForm.md	07.01.2019 16:21	MD-Datei	
Gel_picture.jpeg	07.01.2019 16:21	JPEG-Datei	8
Gel_picture.tiff	07.01.2019 16:21	TIFF-Datei	52
user_inputs.yaml	07.01.2019 16:21	YAML-Datei	

The script will ignore all submission directories without the approvement file.

Script configuration

All relevant options are placed at the beginning of the script file



```
submissions_dir <- file.path("outputs")
raw_files_dir <- file.path("Y:", "CFP", "2018")
summary_dir <- file.path("MQ")
mqpar_template_path <- "mqpar.tmp.xml"

flag_file_name <- "approved.txt"

raw_file_age <- 30 # in days
file_size_filter <- 5000000 # in bytes
barcode_extract_pattern <- "[A-Z]{3}_[A-Z]{2}_\\d{3,5}"
```

Importand settings

Submissions_dir: Directory containing all submissions (output of the Web application)
Raw_files_dir: Directory containing all raw files
Summary_dir: Directory where all summarized resources will be placed
Flag_file_name: file name marking an approved submission (see Submission approvement)
Raw_file_age: raw files exceeding the age will be ignored
File_size_filter: raw files undercutting the size will be ignored