GRAPHICAL USER INTERFACE FOR BIOPROCESS STATES VISUAL ANALYSIS

USER GUIDE

Černý F., Hrnčiřík P., Mareš J., Vovsík J.
Department of Computing and Control Engineering, Institute of Chemical Technology Prague,
Technická 5, 16628 Prague 6, Czech Republic
cernyf@vscht.cz
uprt.vscht.cz

ABSTRACT

The application serves for identification and visualization of a multivariable biochemical processes. The application combines visualization of process variables in time domain with 3D visualization of a process trajectory obtained from the Principal Component Analysis of chosen physiological process variables. The application also can project the process classifications into the plot and it can be used for classification by chosen center points in a process trajectory space too.

KEYWORDS

GUI, PCA, vizualization, classification, biotechnological process, physiological states

1 INTRODUCTION

This application is a robust tool serving as the PCA (Principal Component Analysis) analyst of chosen data, with possibility of visualizes output PCs (Principal Components) interconnected with input data. The visualized input data may but may not be the source data for PCs calculation. The connection between input and output is made by time index.

There is also possible to visualize classification of selected input data also with interconnection with output data. This give as a view of classified PCs according of a prepared input data classification. It's also possible to classify output PCs data by the application and implement gained classification into input data visualization.

2 INSTALLATION

This application is suitable only for the 64bit Windows operation systems for the moment. To run the application the Matlab Compiler Runtime environment (MCR) of version 7.14 need to be installed on the machine.

The application consists from one executable file *PCA_GUI.exe* and doesn't need to be installed. To successfully run the application, the source data file named *AllResults.mat* need to be presented in the same file as the application.

3 SOURCE DATA

The Application input data is stored in the *AllResults.mat* file. This file include variable *AllResults* (thereinafter the *AR*) of the MATLAB Cell class (thereinafter the Cell). The application is distributed with one *AllResults.mat* file which consists from example data to better understand the data structure described next.

To prepare good structured data, the template MATLAB function *PCA_GUI_DataFormation.m* is included. It only prepares data for the application so the function doesn't need to be presented in the same file as the application.

The *AR* has five columns and count of rows (the *I*) is equal to a count of an independent data segments (thereinafter the Formations) (we use one row for each experiment measurement). The *AR* can be described as

$$AR = \begin{Bmatrix} D_1 & FI_1 & DSI & DSP_1 & C_1 \\ \dots & \dots & \dots & \dots & \dots \\ D_i & FI_i & DSI & DSP_i & C_i \\ \dots & \dots & \dots & \dots & \dots \\ D_I & FI_I & DSI & DSP_I & C_I \end{Bmatrix}, \quad (1)$$

where $i \in \langle 1 \; ; \; I \rangle$.

The D_i represents measured and evaluated data defining the process and can be described as

$$D_i = \{VarD_i \quad ValD_i \quad ED_i\}, \tag{2}$$

where $VarD_i$ represents description of variables names by the MATLAB String class (thereinafter the String) and $ValD_i$ represents matrix of variables values. An order of variables names and values must match.

$$VarD_i = \{'var_1' \quad ... \quad 'var_j' \quad ... \quad 'var_J'\}, \tag{3}$$

$$ValD_i = \begin{bmatrix} val_{1,1} & ... & val_{1,j} & ... & val_{1,J} \\ ... & & ... & & ... \\ val_{k_i,1} & & val_{k_i,j} & & val_{k_i,J} \\ ... & & ... & & ... \\ val_{K_i,1} & ... & val_{K_i,j} & ... & val_{K_i,J} \end{bmatrix}, \tag{4}$$

$j \in \langle 1 \; ; \; J \rangle$, $k_i \in \langle 1 \; ; \; K_i \rangle$, J is a count of variables and K_i is a count of measurement points of each variable of selected Formation i. An order of variables names and values must match at each Formation (to allow possibility to join selected Formations in the Application, if some variable is not measured at some Formation, fill the value of that variable by *NaN*). The ED_i represents extra variables, which can be visualized in the Application, but can't be processed by PCA. The ED_i can be described as

$$ED_i = \{VarED_i \quad ValED_i\}, \tag{5}$$

where $VarED_i$ represents extra variables names as the Strings and $ValED_i$ represents matrixes of indexes and values of extra variables

$$VarED_i = \{'evar_1' \quad ... \quad 'evar_{l_i}' \quad ... \quad 'evar_{L_i}'\}, \tag{6}$$

$$ValED_i = \{eval_1 \quad ... \quad eval_{l_i} \quad ... \quad eval_{L_i}\}, \tag{7}$$

$$eval_{l_i} = \begin{bmatrix} indeval_{1,l_i} & eval_{1,l_i} \\ ... & ... \\ indeval_{m_{l_i},l_i} & eval_{m_{l_i},l_i} \\ ... & ... \\ indeval_{M_{l_i},l_i} & eval_{M_{l_i},l_i} \end{bmatrix}, \tag{8}$$

$l_i \in \langle 1 \; ; \; L_i \rangle$, $m_{l_i} \in \langle 1 \; ; \; M_{l_i} \rangle$, L_i is a count of extra variables and M_{l_i} is a count of points of corresponding extra value l_i of corresponding Formation i. Extra variables can have different indexes and count of points.

The D_i is represented in an example source data file *SampleData.mat*. It's a source file for included template *PCA_GUI_DataFormation.m*. It's necessary to prepare a good data structure before run the template.

The FI_i represents description of Formation i and can be described as

$$FI_i = 'formation\_info_i'. \tag{9}$$

The *DSI* is the Cell of a descriptions of sub-formations (thereinafter the Datasets) of Formations and can be described as

$$DSI = \{'set\_info_1' \quad ... \quad 'set\_info_n' \quad ... \quad 'set\_info_N'\}, \tag{10}$$

where $n \in \langle 1 \; ; \; N \rangle$ and N is a count of Datasets. Datasets serves for processing the PCA of chosen Formation across different viewpoints (e.g. across different groups of variables). The *DSI* is the same across all Formations (to allow possibility to join selected Formations in the Application).

The DSP_i is the Cell of Datasets parameters. It can be described as

$$DSP_i = \{DSC \quad ODI_i \quad DV\} . \tag{11}$$

The *DSC* is a count of Datasets. The ODI_i is the Cell of Datasets intervals and it can be described as

$$ODI_i = \{oval_{i,1} \quad ... \quad oval_{i,n} \quad ... \quad oval_{i,N}\} , \tag{12}$$

where $oval_{i,n}$ is a matrix of intervals of the Dataset *n* of Formation *i*, with count of intervals equal to *O*,

$$oval_{i,n} = \begin{bmatrix} begin_{i,n,1} & end_{i,n,1} \\ ... & ... \\ begin_{i,n,o} & end_{i,n,o} \\ ... & ... \\ begin_{i,n,O} & end_{i,n,O} \end{bmatrix} , \tag{13}$$

where $o \in \langle 1 \; ; \; O \rangle$. The *DV* is the Cell of matrixes of Datasets variables

$$DV = \{dval_1 \quad ... \quad dval_n \quad ... \quad dval_N\} , \tag{14}$$

$$dval_n = \lfloor \mathrm{var}_{n,S_1} \quad ... \quad \mathrm{var}_{n,S_p} \quad ... \quad \mathrm{var}_{n,S_P} \rfloor , \tag{15}$$

where $p \in \langle 1 \; ; \; P \rangle$, *P* is a count of selected variables and $S \subset \langle 1 \; ; \; J \rangle$.

The last column of Formations is filled by prepared classification Cells C_i defined as

$$C_i = \{CV_i \quad CVP\} . \tag{16}$$

The CV_i is the Cell of a classification viewpoints definition. The first row is filled by matrixes of classification intervals description and the second row is filled by the Cells of classification description names.

$$CV_i = \begin{Bmatrix} cvval_{i,1} & ... & cvval_{i,q} & ... & cvval_{i,Q} \\ cvident_1 & ... & cvident_q & ... & cvident_Q \end{Bmatrix} , \tag{17}$$

$q \in \langle 1 \; ; \; Q \rangle$ and *Q* is a count of classification viewpoints. Each row in $cvval_{i,g}$ is described by an index of an interval beginning, an index of an interval end and an assignation of a classification (by index corresponding to index of $cvident_q$). That can be described as

$$cvval_{i,q} = \begin{bmatrix} begincv_{i,q,1} & endcv_{i,q,1} & indcv_{i,q,1} \\ ... & ... & ... \\ begincv_{i,q,r} & endcv_{i,q,r} & indcv_{i,q,r} \\ ... & ... & ... \\ begincv_{i,q,R} & endcv_{i,q,R} & indcv_{i,q,R} \end{bmatrix} , \tag{18}$$

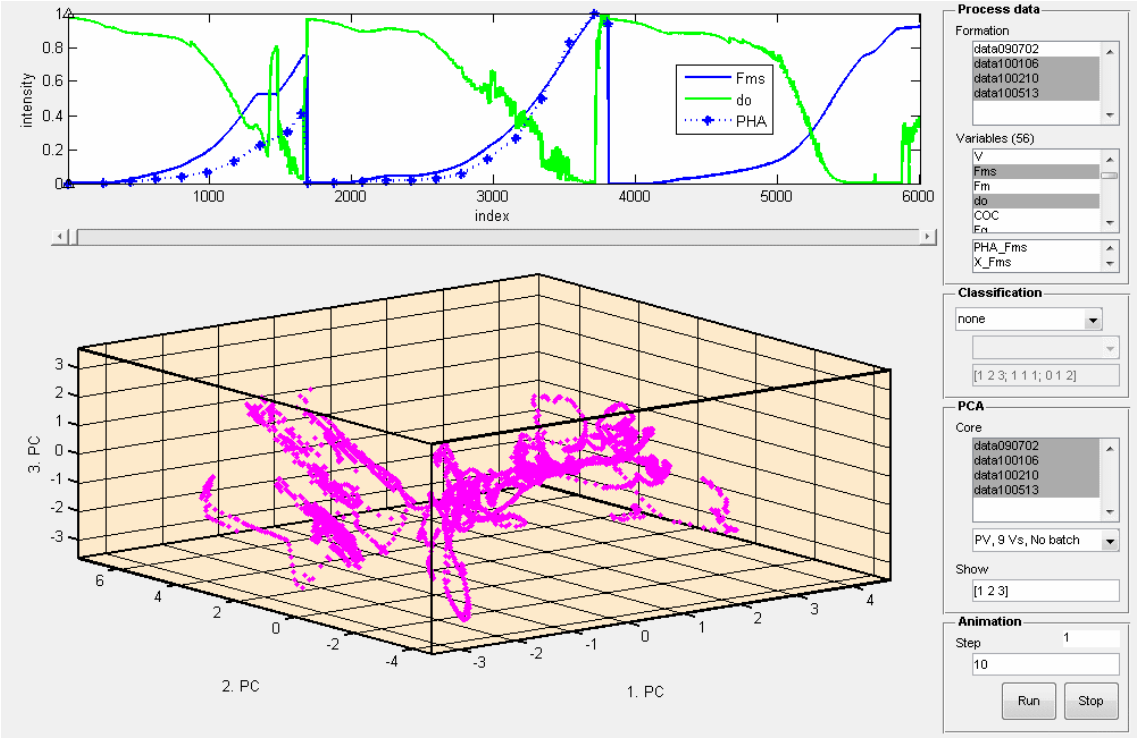
$$cvident_q = \{'ident_{q,1}' \quad ... \quad 'ident_{q,t}' \quad ... \quad 'ident_{q,T}'\} , \tag{19}$$

where $r \in \langle 1 \; ; \; R \rangle$, $t \in \langle 1 \; ; \; T \rangle$, *R* is a count of continuous classification intervals and *T* is a count of corresponding classification viewpoint types. The *CVP* is the Cell of classification viewpoints names

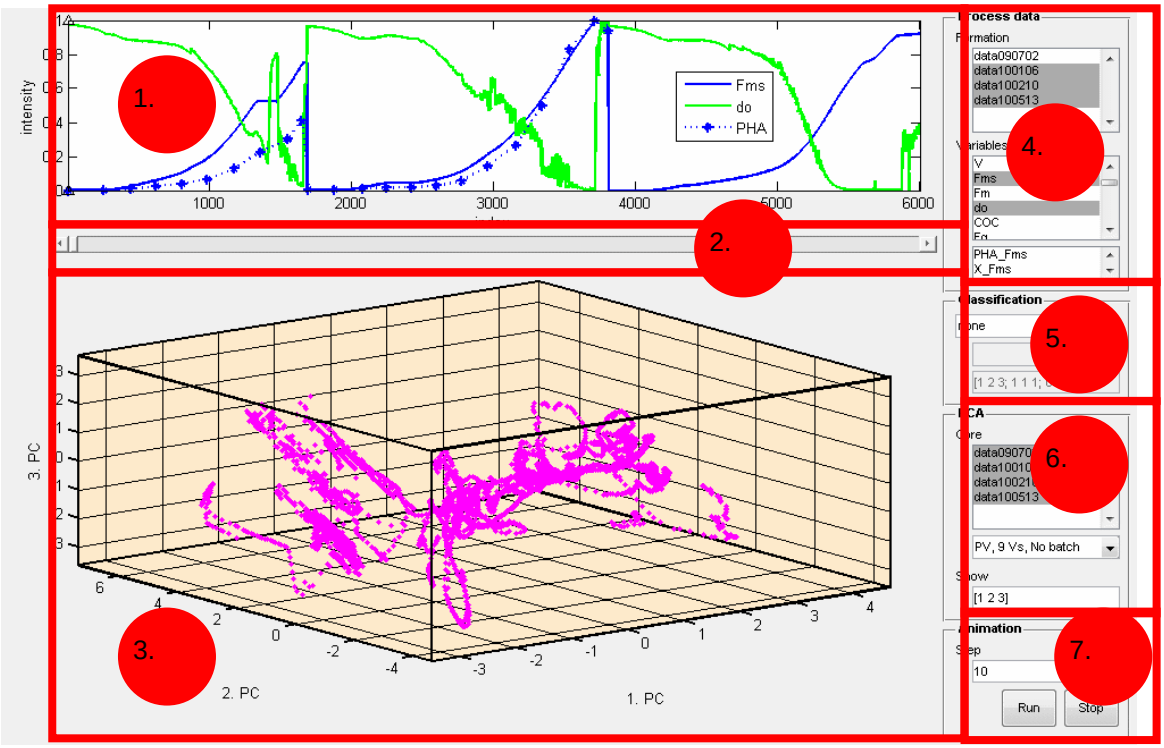
$$CVP = \{'name_1' \quad ... \quad 'name_q' \quad ... \quad 'name_Q'\} . \tag{20}$$

4 THE APPLICATION GUI

Picture (1) – PCA Application GUI



Picture (2) – PCA Application GUI – parts



The application GUI has seven main parts. The 1st and 3rd part serve as application output and others parts serve as an user control.

The 1st part serves for input data visualization.

The 2nd part represents the horizontal scroll bar for time index selection. It connects input and output data visualization across that time index.

The 3rd part serves as the principal components (PCs) visualization.

The 4th part includes selection of one or more Formations and selection of Variables to plot at the 1st part (the extras variables lie at the bottom of this part).

The 5th part includes user controls for a classification selection. User can chose type of classification between none, prepared, nearest and distance. None means that data will not be classified. It will cause a color differentiation of output plot data before and after selected time index. Prepared allows user to chose between already prepared classification (included in the application data source). Nearest enables user to insert a centers definition to classify data according these centers. Data will be classified to the nearest center. Centers is defined by a matrix with a count of rows equal to a count of centers and a count of columns equal to a dimension of classification. Data will be classified by PCs variables inserted in the 6th part at a field Show, so a maximal dimension size is depended on a count of PCs at this field. Distance enables a user to insert a centers definition and a maximal distance to these centers to classify data according these areas. Data will be classified to the center, if it is in a range. This classification is defined the same way as the nearest but has one additional column with maximal distance values.

The 6th part serves as a PCA user control. The Core field serves to choose Formations for PCA core calculation. The next field serves to Dataset selection. The bottom field serves for PCs selection. It is a one row matrix with at least three columns. The first three PCs will be visualized at the output visualization part. The more columns are there, the higher dimension of the nearest and the distance classification is possible (but maximal equal)

The 7th part serves to user control of an animation of the application. It's possible to change an animation step and to run and to stop an animation.

Picture (3) – PCA Application GUI - toolbar

The GUI's toolbar contains a few standard MATLAB Figure (thereinafter the Figure) options as rotate plot in 3D, data cursor and print figure order. There are also three additional buttons. The *STAT GIF* button causes a saving of an actual GUI's screen into a *.gif file format. The *DIN GIF* button serves for an animation *.gif file format creation. This button has to be toggled before the order button *Run* is pushed. The *F* button creates the new Figure and fills it with information from the input data visualization part (except of a classification and a time index mark).

5 ACKNOWLEDGEMENT

Development of this application was supported by the 6th Framework Programme of the European Community under contract No. NMP2-CT-2007-026515 "Bioproduction project - Sustainble Microbial and Biocatalytic Production of Advanced Functional Materials" and by the fund No. MSM 6046137306 of Ministry of Education of Czech Republic. This support is very gratefully acknowledged