

User's Guide to the BREReA Program

Contents

I	Background Removal	3
1	Background removal program: Creation of user profile	3
1.1	Input of general parameters	3
1.2	Input of protein-specific parameters	4
2	Program Usage: Brief User guide	6
2.1	Creation of user profile	6
2.2	Method implementation	7
3	Background Removal: Input and Output	8
3.1	Input of the Data	8
3.2	Output of results	9
4	Algorithm	12
4.1	Detection of Nonexpressing Areas	12
4.2	Extraction of Points of Support	12
4.3	Approximation of background signal	12
4.4	Background removal	13
4.5	Special Cases	13
4.6	Notations of proteins	14
II	Registration and Averaging	15
5	Creation of user profile	15
5.1	Input and output directories	15
5.2	Input of program parameters	16
6	Program Usage	18
6.1	Construction of preliminary GCP template and reference pattern	18
6.2	Registration of dataset and Update of template and reference data	18
6.3	Registration of an individual pattern	19
6.4	Update of induced reference data	20
6.5	Output files	20
6.6	Interactive Graphic Interface	21
7	Program Implementation: Brief User guide	22

8	Program Implementation: Detailed Description	24
8.1	Input of the data	24
8.2	Creation of preliminary template and reference pattern	25
8.2.1	Input of the GCP template	26
8.2.2	Input of the reference pattern	26
8.3	Spline Approximation	26
8.3.1	Initial values of knots	27
8.3.2	Approximation procedure	27
8.3.3	Error diagnostic	28
8.4	Feature Extraction by Spline Method	28
8.4.1	Input of features	28
8.4.2	Standardization of the extracted features	29
8.4.3	Standard features.	30
8.4.4	Ground control points	30
8.4.5	Output results	30
8.5	Wavelet decomposition and Feature Extraction	31
8.5.1	Fast redundant dyadic wavelet transform	31
8.5.2	Feature extraction: Zeros of high pass	31
8.5.3	Standardization of the extracted features	32
8.5.4	Output of results	32
8.6	Registration and Output	33
8.7	Update of GCP template and reference pattern	33
8.7.1	GCP Averaging	34
8.7.2	Data averaging	34
8.7.3	Induced registration	34
8.7.4	Output of results	34

Introduction

BreReA is a software designed to standardize the data on segmentation gene expression in *Drosophila*. The quantitative data are extracted from confocal images as described in (Janssens, 2005). To construct reference data or a spatiotemporal map of expression of all segmentation genes it is necessary to combine data from many individuals scanned in different experiments. The data integration is impossible without bringing the data to the standard form with zero background and free of positional variability. BreReA implements background removal, registration and averaging methods.

The background removal program is designed to standardize the gene expression data obtained in different experiments and stained using different antibodies. The methods for immunofluorescent labeling of biological objects in situ give rise to a low level of nonspecific staining, or background. The background removal method is based on observation that background can be well fit by a very broad two dimensional paraboloid. The paraboloid is determined from the areas of embryos in which a given gene is not expressed and the whole pattern is then normalized by this paraboloid to remove background from the entire embryo (Myasnikova et al., 2005).

The registration method is designed to register one-dimensional gene expression patterns of *Drosophila* segmentation genes stained in embryos at the same stage of development. The

expression patterns cannot be directly superimposed because of small individual differences among the population. Finding of the coordinate transformations such that expression domains of the gene product in different embryos would be superimposed defines the registration problem. If several genes are stained in the same set of embryos the registration of one of these genes induces the registration of all the others.

The registration is implemented by means of the point mapping technique (Brown,92). It is based on the extraction of “ground control points” (GCP), which are a small number of characteristic features in each pattern, and application of the affine coordinate transformation to make the patterns coincide as closely as possible. There are two options to extract discrete characteristic features from gene expression patterns: by means of quadratic spline approximation or wavelet decomposition (FRDWT). The detailed description of the algorithms is given in (Myasnikova et al.,2001, Surkova et al., 2008). The expression patterns of most segmentation genes are made up of several expression domains, each of which has at least one concentration maximum. It is natural to take the maxima and/or minima of one-dimensional gene expression pattern as GCPs for registration.

Program usage: `qdata-gui`

The program opens the main menu page. To run the program a user has to construct a user profile or use a profile constructed during previous sessions. These profiles can be selected from the list of Available profiles. BREReA opens the **Open profile** menu containing the list of existing profiles. The user can either select an existing profile or to choose the 'default' item and create a new profile (see Fig.1). On the first run the menu only contains one item 'default'.To create a new profile choose the **Go to Background removal** or **Go to Registration and Averaging option**.

Part I

Background Removal

1 Background removal program: Creation of user profile

Define input data and output directories. To define the directory with input data open the **Input Open File Dialogue** and choose the directory with data files. The names of data files, `<filename>`, are composed of the path and the `<basename>` with the extension(s). The `<basename>` is automatically extracted from the `<filename>` (example: `<filename>:=/home/data/BCE/ab1.proc.d.asc` then `<basename>:=ab1`). The input data file has the structure described in 3.1.

To define the output directory, `<output_dir>`, open the **Output Open File Dialogue** and choose the directory where the output files will be saved. The names and structure of output files are described in ??.

Press the **OK** button to open the list of input files. Choose the file of interest from the list using the mouse. The 1D patterns of genes stained in the embryo will appear. There can be from one to three genes stained in one embryo.

1.1 Input of general parameters

The general parameters are common for all the patterns.

Input the range of AP-values of nucleus coordinates that will be considered in background estimation using the **Xmin** and **Xmax** spin buttons. This is necessary to exclude the out-of-focus areas at both the anterior and posterior tips of the embryo (5% of embryo length). Input the range of DV-values that will be considered in the background estimation using the **Ymin** and **Ymax** spin buttons. In the **Stripe Width** text field the area of the embryo in DV-direction is defined that will be rescaled by the estimated background and included into the output file. This area should not necessarily coincide with the area defined by the **Ymin** and **Ymax** spin buttons. The background can be estimated, for example, from the 50% central strip and then applied to the whole embryo or 80% strip, excluding 10% from dorsal and ventral sides.

The input parameters are:

Xmin: lower limit of the AP-coordinate. By default is 0;

Xmax: upper limit of the AP-coordinate. By default is 100;

Ymin: lower limit of the DV-coordinate. By default is 45;

Ymax: upper limit of the DV-coordinate. By default is 55;

Stripe Width: the percent of the embryo in DV-direction that will be rescaled by the estimated background and included into the output `<basename>.norm` file. By default is 80%;

1.2 Input of protein-specific parameters

The protein specific parameters are set for each pattern separately through the text fields located on the right of the corresponding graph. By default the background removal method will be applied to all the patterns. To exclude any pattern from processing uncheck the corresponding **Yes** checkbox.

First, input the gene identifiers using the text field:

1 - one letter code: `<gcode>`, lowercase one-letter identifier of the gene (e.g., e for *eve*).

The notations used for Drosophila segmentation genes are given in 4.6;

Parameters controlling non-expressing areas. There are two ways to define non-expressing areas. The approximate positions of non-expressing areas for all the segmentation genes are predetermined and stored in built-in files `<INSTDIR>/share/qdata-gui/bgremdata/prot.<gcode>`. To load the predefined borders of non-expressing areas for a given gene press the **Find** button. The program will open 3 spin buttons for each area:

Left border: x_k^{lft} , the AP-coordinate of the left border of the *kth* non-expressing area;

Right border: x_k^{rht} , the AP-coordinate of the right border of the *kth* non-expressing area;

Intensity threshold: ρ_k , parameter is used to define the points of support for the background estimation. These points are selected from non-expressing areas if their deviation from the mean intensity value in the area doesn't exceed ρ_k ;

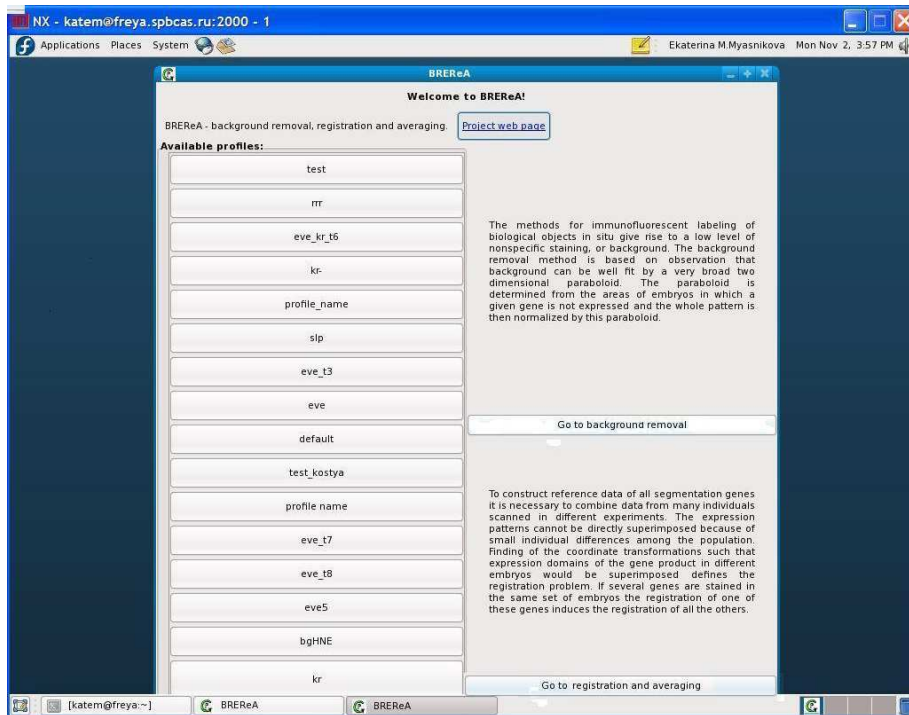


Figure 1: Screenshot of the BREReA main page. To run the program a user has to construct a user profile or use a profile constructed during previous sessions. These profiles can be selected from the list of Available profiles. To create a new profile choose the Go to Background removal or Go to Registration and Averaging option.

To directly define the areas press the **Add** button and input the parameter values. Press the button as many times as the number of non-expressing areas. To delete a non-expressing area press the **Delete** button.

The value of **Intensity Threshold**, ρ_k , is chosen to permit the extraction of a sufficient number of points of support to construct an initial approximation of background. Experimentally it was found that it is reasonable to take the parameter value approximately equal to $2/3$ of the range of expression values in the area. This value allows to remove any expressing nuclei that may exist in the area. Each region must be narrow enough so that the background does not vary significantly within it; thus a large non-expressing region will be subdivided into several ones. The input value of ρ_k is automatically corrected by the program.

The non-expressing areas outlined by brown frames appear on the graphs (see fig.2). All the parameters can be edited at any stage of the program implementation. The new parameter values are saved for each protein in the files `<output_dir>/<basename>.<gcode>`. To restore initial values of parameters press the **Copy** button.

Parameters controlling the shape of background. The shape of background can be selected from the **Mode** drop-down list. The most general background is parabolic, all the rest types are selected if there is not enough information to estimate the background from the defined non-expressing areas. It is important to keep in mind that these types

of background are only used to set the initial values of paraboloid parameters and will be refined during the background optimization. As a result the background will take a general parabolic shape. For more detailed description see 4.5.

parabolic: The most general case. Background is estimated by a 2D paraboloid $S(x, y) = \alpha_{11}x^2 + \alpha_{22}y^2 + \alpha_{12}xy + \alpha_1x + \alpha_2y + \alpha_0$ from the area limited by **Xmin**, **Xmin**, **Ymin** and **Ymax**.

symmetry in DV: Background is assumed symmetric in y direction and estimated as $S(x, y) = \alpha_{11}x^2 + \alpha_{11}y^2 + \alpha_1x + \alpha_1y + \alpha_0$.

flat in AP: The paraboloid is estimated as $S(x, y) = \alpha_{22}y^2 + \alpha_1x + \alpha_2y + \alpha_0$.

constant in AP: The paraboloid in is estimated as $S(x, y) = \alpha_{22}y^2 + \alpha_2y + \alpha_0$.

constant in DV: The paraboloid is estimated as $S(x, y) = \alpha_{11}x^2 + \alpha_1x + \alpha_0$.

The symmetric shape of background is set if the **Symmetry in AP** checkbox is checked. If, for example, the non-expressing areas are all located in either the anterior or posterior part of the embryo there is no information about the background in the opposite part and it is estimated as symmetric in the AP direction.

2 Program Usage: Brief User guide

In this section a very brief user guide is given. The more detailed presentation of the method is described in the 4 section.

2.1 Creation of user profile

Selection of input and output directories – Select input and output directories from the **Input** and **Output** file selection dialogues, and press the **OK** button to open the list of input files. Choose the file of interest from the list using the mouse. Three 1D patterns of genes stained in the embryo will appear.

Input of general parameters – Use the **Xmin** and **Xmax** text fields to set the range of AP-values that will be considered in background estimation. This is necessary to exclude the out-of-focus areas at both the anterior and posterior tips of the embryo (5% of embryo length). Use the **Ymin** and **Ymax** spin buttons. to define the range of DV-values that will be considered in the background estimation. In the **Stripe Width** text field the area of the embryo in DV-direction is defined that will be rescaled by the estimated background and included into the output file. This area should not necessarily coincide with the area defined by the **Ymin** and **Ymax** spin buttons. The background can be estimated, for example, from the 50% central strip and then applied to the whole embryo or 80% strip, excluding 10% from dorsal and ventral sides.

Input of the pattern-specific parameters – By default the background removal method will be applied to all the patterns. To exclude any pattern from processing uncheck the corresponding **Yes** checkbox. Use the text field named as **1-letter code for gene name** to name each of the three expression patterns according to the common notation of the corresponding gene name. For notations see 4.6.

Definition of non-expressing areas – the approximate positions of non-expressing areas for all the segmentation genes are predetermined and stored in built-in files. To load the predefined borders of non-expressing areas for a given gene press the **Find** button. The non-expressing areas outlined by brown frames appear on the graphs. The AP-coordinates of the borders of these areas are shown in the **Left border** and **Right border** spin buttons for each pattern. The borders of the non-expressing areas can be edited by the user. For each non-expressing area the **Intensity Threshold** parameter is also set. This parameter is used to define the points of support for the background estimation. These points are selected from non-expressing areas if their deviation from the mean intensity value in the area doesn't exceed the **Intensity Threshold**. This parameter can be also edited. The unnecessary non-expressing areas can be deleted by pressing the **Delete** button. Extra areas can be added by pressing the **Add** button.

Definition of background shape – the shape of background can be selected from the **Mode** drop-down list. The most general background is parabolic, all the rest types are selected if there is not enough information to estimate the background from the defined non-expressing areas. The symmetric shape of background is set if the **Symmetry in AP** checkbox is checked. For more details see 4.5.

To save the profile select the **Save** or **Save as** option from the **Profile** drop-down menu and input the profile name. To quit the program select the **Quit** option from the menu. To open the main menu choose the **Starter** option.

2.2 Method implementation

Refresh button – the program extracts the set of points of support from nonexpressing areas. We find a nucleus with the minimal expression for each nonexpressing area and then pick all the nuclei with expression not exceeding the minimal value by a given value of **Intensity Threshold**. The program finds the initial values of the coefficients of paraboloid providing the best fit to the set of points of support by LS method. The results are visualized (see fig. 2).

Apply button The background paraboloid is corrected by the iteration method. At each iteration the set of points of support is supplemented with those nuclei, if any, not belonging to nonexpressing areas but where expression is lower than the current background. The results are visualized (see fig. 3).

Save button – Finally, background is removed from the the area of the embryo defined in **Stripe Width** text field entire embryo by a linear mapping of intensity that transforms fluorescence at or below background level to zero and transforms maximum fluorescence (255) to itself. The normalized patterns are saved in the file `<output_dir>/<basename>.norm`. The information about normalization is put into the header (See Program Usage). The normalization coefficients α_{ij} are saved into the file `<output_dir>/<basename>.norm_coefs`. The new protein-specific parameters are saved into the files `<output_dir>/<basename>.<gcode>`. The results are visualized (see fig. 4).

Copy button restores the initial predefined parameter values from the `<INSTDIR>/share/qdata-gui/bgremdata//<basename>.<gcode>` files.

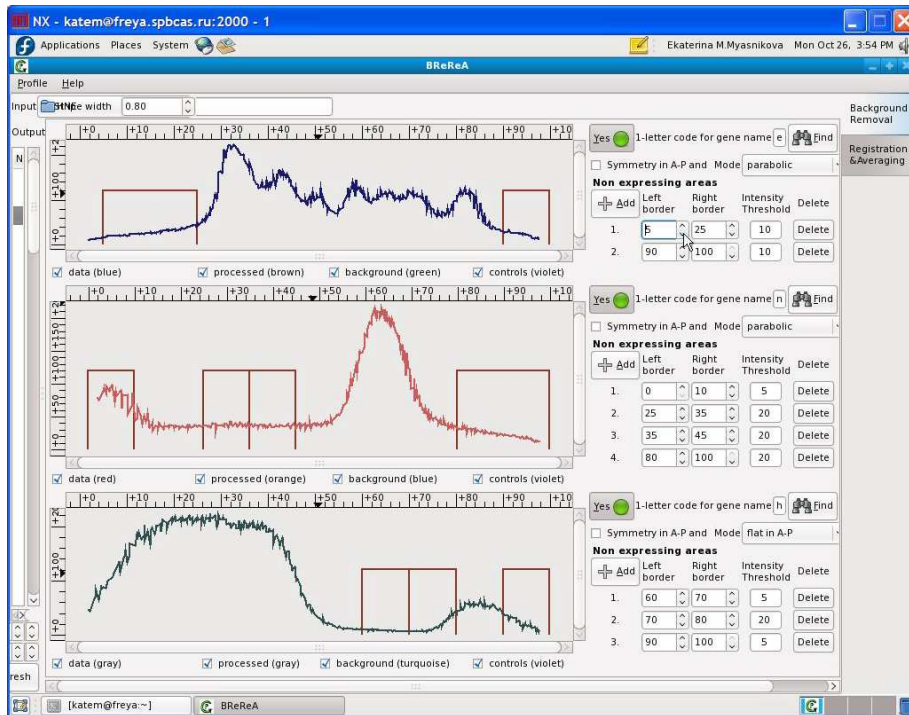


Figure 2: Background removal program. Definition of non-expressing areas. The non-expressing areas outlined by brown frames appear on the graphs. The AP-coordinates of the borders of these areas are shown in the Left border and Right border text fields for each pattern.

If the quality of background estimation is non-satisfactory, the background can be corrected at any stage of the procedure. Input the corrected values of parameters for non-expressing areas and/or background shape and press the **Refresh** and **Apply** buttons again. Repeat until the satisfactory results are achieved. The corrected parameter values are saved into the output directory and will be used for processing of next embryos. To restore the initial predefined parameter values press the **Copy** button.

3 Background Removal: Input and Output

3.1 Input of the Data

Input of 1, 2 or 3 expression patterns of <protein_i> from the data file <filename>. The data are extracted from the <stripe_width>% central strip. The file may contain data from 1, 2 or 3 channels.

Structure of the data file:

```
*****
# comment lines
...
nucleus # x-coord y-coord 1 channel data [optional] 2 channel data [optional] 3 channel data
*****
```

Example of a data file:



Figure 3: Background estimation. The points of support are extracted from the non-expressing areas and the initial value of background is estimated by fitting to these points. To see the initial approximating parabola on the graphs press the Refresh button. To refine the background shape press the Apply button.

```
# comment lines
# comment lines
0 0.854948 42.537400 82.001731 0.736363 15.068856
1 1.172000 44.804700 77.090513 4.746355 13.092463
2 1.759100 47.222200 108.009691 2.34899 14.592068
3 2.020420 39.294100 114.663652 0.099332 21.335432
```

The data are read off into the following arrays:

x_j - x -coordinate of j th nucleus
 y_j - y -coordinate of j th nucleus
 C_{ij} - expression of <protein_i> in j th nucleus
 $j = 1, \dots, N$

3.2 Output of results

The output files are saved into the <output_dir> directory.

<basename>.norm – The normalized pattern (expression values are transformed). The header contains information about the background removal enclosed in tags;

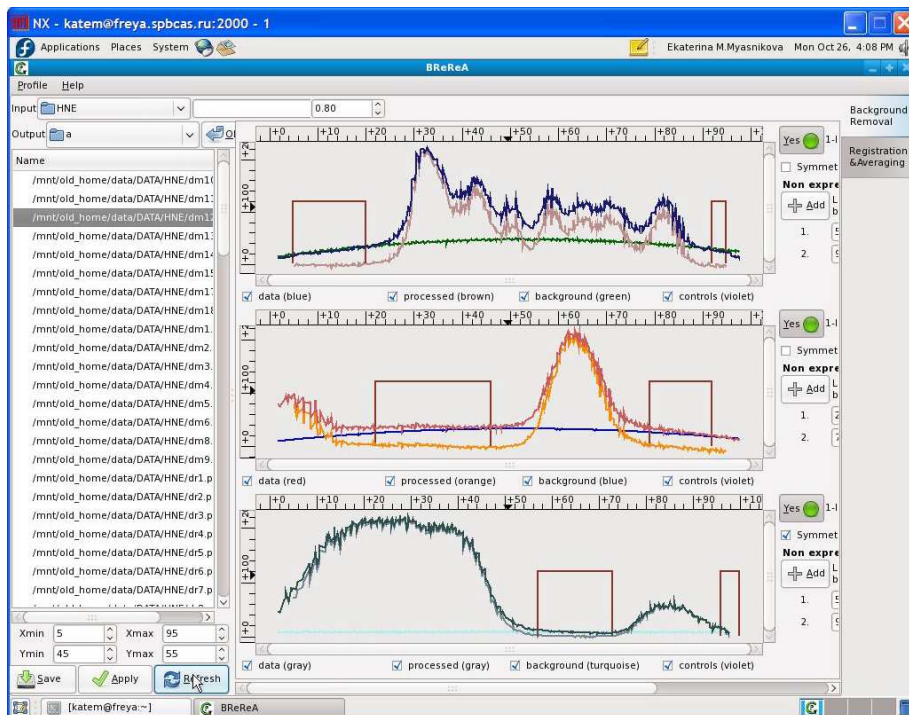


Figure 4: Normalization of expression patterns. To normalize the patterns by the estimated background press the Save button. The rescaled patterns (with removed background) appear on the graphs and are saved into output files.

`<basename>.coefs` – Contains normalization parameters.

`prot.<protein_i>` – contains protein-specific parameters controlling normalization of `<protein_i>`. The parameters are either input or revised by the user.

Output of the normalized patterns

All the normalized patterns are written into the file `<output_dir>/<basename>.norm`. The file has the same structure as the input file.

The header contains information from BREReA enclosed in tags: `<bgrem>` and `<\bgrem>`. Version string of the code enclosed in tags `<version>` and `<\version>` together with modification date; mode of background estimation enclosed in tags: `case` and `<\case>`; proteins stained in the embryo enclosed in tags: `<channel1>` and `<\channel1>`, `<channel2>` and `<\channel2>`, `<channel3>` and `<\channel3>`; coefficients of background paraboloid enclosed in tags: `<protein1 paraboloid coefficients>` and `<\protein1 paraboloid coefficients>`. They are: α_{11} , α_{22} , α_1 , α_2 , α_0 , α_{12} . Paraboloid coefficients for `<protein2>` and `<protein2>` are given in the same format.

Structure of the file:

```
*****
HEADER:
<bgrem>
<version> 2.1 <\version>
case 0 <\case>
```

```
<channel1> even-skipped <\channel1> <channel2> runt <\channel2> <channel3>
fushi-tarazu <\channel3>
```

```
<protein1 paraboloid coefficients>
-0.005853 -0.005859 0.559132 0.513687 5.941217 0.001116
<\protein1 paraboloid coefficients>
<protein2 paraboloid coefficients>
-0.005078 -0.006162 0.547560 0.620668 0.031140 -0.000250
<\protein2 paraboloid coefficients>
<protein3 paraboloid coefficients>
-0.004014 -0.004994 0.436228 0.497305 -4.979109 -0.000115
<\protein3 paraboloid coefficients>
<\bgrem>
```

FILE STRUCTURE:

#	<i>x</i> coord	<i>y</i> coord	normalized protein1	normalized protein2 [optional]	normalized protein3 [optional]
0	0.854948	42.537400	82.001731	-1.863357	15.068856
1	1.172000	44.804700	77.090513	3.735211	13.092463
2	1.759100	47.222200	108.009691	-2.541100	14.592068
3	2.020420	39.294100	114.663652	-0.654790	21.335432

...

Output of the protein-specific parameters

Example of prot.<protein_i> file: (prot.e)

```
# variant /uu/
0
# symmetry /sym/
1
# domains /n_dom x_lft[0] x_rht[0] x_lft[1] x_rht[1] ...
2 6.000000 16.000000 90.000000 95.000000
# max range /rho/
20.000000 20.000000
```

Output of normalization coefficients

Structure of the file:

```
<protein1>  $\alpha_{11}$   $\alpha_{22}$   $\alpha_1$   $\alpha_2$   $\alpha_0$   $\alpha_{12}$ 
<protein2>  $\alpha_{11}$   $\alpha_{22}$   $\alpha_1$   $\alpha_2$   $\alpha_0$   $\alpha_{12}$ 
<protein3>  $\alpha_{11}$   $\alpha_{22}$   $\alpha_1$   $\alpha_2$   $\alpha_0$   $\alpha_{12}$ 
```

Example of the file:

```
e -0.005853 -0.005859 0.559132 0.513687 5.941217 0.001116
r -0.005078 -0.006162 0.547560 0.620668 0.031140 -0.000250
f -0.004014 -0.004994 0.436228 0.497305 -4.979109 -0.000115
```

Below the detailed presentation of the methods is given.

4 Algorithm

4.1 Detection of Nonexpressing Areas

Aim: This module defines the set of nonexpressing areas by setting their borders in A-P direction.

Before calculating the background profile from a given embryo, it is necessary to eliminate out of focus nuclei from the edges. This problem is solved by histogram analysis. To cut off the nuclei with residual fluorescence a threshold value is selected from the histogram, and nuclei with expression lower than this value are excluded from the expression pattern. The threshold value is set equal to

To make it easier to determine the nonexpressing regions in two dimensions we straighten the curved stripes. This is done by means of a continuous and differentiable coordinate transformation $x \rightarrow x'$ which maps nearby nuclei with similar values of gene expression to points with equal x' coordinates while keeping their y coordinates unchanged. We apply a single transformation to all embryos. The method for stripe straightening is given in Appendix.

This transformation of an expression pattern allows us to treat nonexpressing areas as functions of x' only, and to consider each such area as a vertical strip on the embryo body.

The number and borders of nonexpressing areas are input from the `prot.<protein_i>` file. These parameters are set for the central 10% strip along x -axis. It may happen that for a given pattern one or more of predefined nonexpressing areas don't intersect with any actual nonexpressing area in this embryo. In such a case this area should be excluded. To do this we find for each predefined area a nucleus with the minimal value of expression. Then we check whether the minimal values in domains increase towards the center of an embryo in the anterior half and decrease in the posterior half. If it is not true for one or more of minimal values the corresponding domain is excluded. Thus we have a set of K_{non} nonexpressing areas I_k defined by their left and right borders x_k^{lft} , x_k^{rht} , $k = 1, \dots, K_{non}$.

4.2 Extraction of Points of Support

Aim: to extract the points of support from the nonexpressing areas.

The background is approximated by a quadratic paraboloid fit to the points of support. The initial guess of points of support is extracted from the straightened nonexpressing regions of the two-dimensional pattern as follows. The expression pattern of each gene is composed of N nuclei characterized by x_i , y_i and C_i , i.e., the x position, y position, and fluorescence levels of nucleus i respectively. We consider only one fluorescence channel because the procedure is applied to one channel at a time. We pick all the nuclei from each region I_k with expression not exceeding a given value ρ_k . The value of ρ_k is defined for each I_k and input from the parameter file `prot.<protein_i>`. The set of points of support selected from the region I_k is denoted as \mathbf{U}_k and the total set of points of support is $\mathbf{U} = \bigcup_k \mathbf{U}_k$.

4.3 Approximation of background signal

The approximating paraboloid $S(x, y) = \alpha_{11}x^2 + \alpha_{22}y^2 + \alpha_{12}xy + \alpha_1x + \alpha_2y + \alpha_0$ used in this method is found by an iterative optimization procedure. The initial values of the

coefficients are obtained by least squares. We minimize

$$L_0 = \sum_{\mathbf{U}} [a_i - S^{(0)}(x_i, y_i)]^2 \quad (1)$$

with respect to the coefficients of the initial paraboloid $S^{(0)}(x, y)$.

At each iteration the set of points of support \mathbf{U} is supplemented with those nuclei, if any, not belonging to nonexpressing areas but where expression is lower than the current background, $\mathbf{U}_-^{(m)}$, and the next vector of paraboloid coefficients is found by minimizing

$$L_{m+1} = \sum_{\mathbf{U} \cup \mathbf{U}_-^{(m)}} [a_i - S^{(m+1)}(x_i, y_i)]^2, \quad (2)$$

where $S^{(m)}$ is the paraboloid obtained at the m th step.

The iteration procedure is converged when either there are no points with expression lower than the background or when addition of such points to the set of points of support does not decrease the value of L_{m+1} .

4.4 Background removal

Finally, background is removed from the entire embryo by a linear mapping of intensity that transforms fluorescence at or below background level to zero and transforms maximum fluorescence (255) to itself. The transformation is given by

$$a_i^{norm} = \max \left(\frac{a_i - S(x_i, y_i)}{255 - S(x_i, y_i)} 255, 0 \right), \quad i = 1 \dots N, \quad (3)$$

which yields a normalized expression pattern in which each nucleus is characterized by the vector $\{x_i, y_i, a_i^{norm}\}$, $i = 1, \dots, N$.

4.5 Special Cases

Degenerate cases of background shape

symmetry in DV: Due to the ellipsoidal shape of an embryo the nonexpressing areas which are located in the 20% anterior and posterior portions of an embryo cover not more than central 20% of an embryo width. It is evidently not enough to make any conclusions about the structure of the background in D-V direction. At the same time if all the points of support are extracted only from the narrow central part of an embryo, the anterior and posterior parts of paraboloid cannot be accurately approximated. The background in this situation is assumed symmetric in y direction. Under this assumption the approximation is performed by a quadratic curve fit to the nonexpressing areas extracted from 10% strip along A-P axis. The background is approximated only in x direction by the parabola $\alpha_{11}x^2 + \alpha_1x + \alpha_0$ and then the symmetric 2D paraboloid is restored as $S(x, y) = \alpha_{11}x^2 + \alpha_{11}y^2 + \alpha_1x + \alpha_1y + \alpha_0$.

flat in AP: If the nonexpressing areas don't provide us with any information about the quadratic curvature in A-P direction the attempt to approximate the background by a paraboloid in the general form may lead to the overestimated background or even the "overtuned" (convex) parabola in x direction (i.e., $\alpha_{11} > 0$). Even if this option

is not used as an input parameter it will be automatically set by the program in case if the optimal paraboloid estimation gives $\alpha_{11} > 0$. Usually it happens in case of normalization of early Hb. The paraboloid is estimated as $\alpha_{22}y^2 + \alpha_1x + \alpha_2y + \alpha_0$.

constant in AP: The degenerate case when in the nonexpressing areas there is no information about curvature in x direction and non-expressing areas cover a small part of the embryo length. Usually it happens in case of normalization of Hb or early patterns. The paraboloid in this case is estimated as $\alpha_{22}y^2 + \alpha_2y + \alpha_0$.

constant in DV: The degenerate case when in the nonexpressing areas there is no information about DV direction. Usually it happens when the embryo is deformed by the cover glass. The paraboloid in this case is estimated as $\alpha_{11}x^2 + \alpha_1x + \alpha_0$.

Bicoid exponenta *bcd* is a special case because its posterior boundary of expression is indefinite, requiring further steps beyond the symmetry assumption. It is well known that Bcd protein is distributed in an exponential concentration gradient with a maximum at the anterior tip, reaching background levels somewhere in the posterior third of the embryo. We normalize the *bcd* pattern by requiring that the normalized pattern have an exponential profile. The algorithm for background removal described above does not always yield an exponential profile when applied to the *bcd* pattern, and hence requires further correction. This correction is performed by fitting the one-dimensional pattern extracted from the central 10% strip by $a^{norm} = A \exp(-x/\lambda) + B$, where a^{norm} is normalized intensity given by (3), and B is a constant correction to the paraboloid background.

Symmetry assumptions The symmetric shape of background is set if the 'Symmetry in AP' checkbox is checked. The nonexpressing areas of some patterns are not large enough to provide a set of points of support sufficient to construct an appropriate background paraboloid. The symmetry assumption is required for proteins with very broad domains, such as *hb*, *cad* and some pair-rule genes at early stages. It works as follows. Because the paraboloid approximating expression in a mutant embryo is very close to being symmetric, it is reasonable to double the initial number of points of support \mathbf{U} by assuming them to be symmetric in the x direction. We do this by inserting copies of the points at symmetric positions with x coordinates $100 - x_i$. When nonexpressing areas are very small, the symmetry assumption improves results. This is particularly important for the maternal genes *cad* and *bcd*, as well as the late pattern of the gap gene *hb*. The nonexpressing regions for these expression patterns comprise only 10-15% of the length of the embryo.

4.6 Notations of proteins

The list of proteins/RNA for which there exist parameter files:

- a** – RNA;
- b** – Bicoid;
- c** – Caudal;
- e** – Even-skipped;
- f** – Fushi-tarazu
- g** – Giant;
- h** – Hunchback;
- i** – Hairy;

k – Kruppel;
n – Knirps;
o – Odd-skipped;
p – Paired;
r – Runt;
s – Sloppy-paired;
t – Tailless.

Part II

Registration and Averaging

5 Creation of user profile

The user can either select an existing profile from the BREReA main page or create a new profile (see Fig.1). To create a new profile the user defines input and output parameters in the **Data**, **Reference**, **FRDWT** and **Spline**, windows. At the next run of the program the existing profile can be edited and saved.

Note!! The profile is built for each gene and for each time class separately, even if several genes are stained in one and the same group of individuals. Each profile is created to only construct one reference pattern regardless of whether the direct or the induced registration is used.

5.1 Input and output directories

Data window

Define input data and output directories. Each input data directory should contain the files with the gene stained in the same microscope channel. The name of this directory will be considered as an identifier of the data group: **group name**. For example, if the input directory is `home/data/FRE`, then `<group_name>` is `FRE`.

Inputs:

name: `<data_dir>`, choose the list of directories with input data;

filter: `<filter>`, for each input directory define the extension of data files (typically, `.norm` or `.proc.d.asc`);

channel: `<channel>`, for each input directory define the channel with the data to be registered.

To add the directory to the list press the **Add** button.

Output:

name: `<output_dir>`, choose the output directory.

To save the output directory press the **Apply** button.

To save different types of output files the program creates the following directories:

<output_dir>: output directory defined by the user (e.g., `home/output`);
<output_group>= `<output_dir>/<group_name>` (e.g., `home/output/FRE`);
<output_tclass>= `<output_dir>/<group_name>/T<tclass>` (e.g.,
`home/FRE/output/T8`);

5.2 Input of program parameters

The program input parameters are defined in the **Reference**, **FRDWT**, **Spline** and **Figures** windows.

Reference window

Define parameters to create GCP template and reference data. Gene and timeclass identifiers are used to create names of output directories and files containing the feature extraction and registration results. A screenshot of the **Reference** window is presented in Fig. 6.

Path: `<output_ref>`, directory to save GCP template and reference data. By default coincides with the output directory defined in the **Data** window;

One letter code L: `<gcode>`, lowercase one-letter identifier of the gene (e.g., `e` for *eve*);

Time class ID: `<tclass>`, one-digit identifier of the time class (or any user-defined group of embryos);

Method: `<method>`, the method for GCP extraction, **spline** or **FRDWT**, is selected using the dropdown menu. These two methods provide very similar results in most cases, but it is generally recommended to apply spline approximation to broad expression domains, while for narrow sharp stripes the FRDWT method is preferable;

Number of GCPs: `<nGCP>`, number of GCPs to be used in registration. To define this parameter it is necessary to thoroughly inspect all the individual expression patterns at a given stage of development and select the most stable and well-localized extremal features;

Bottom border: lower limit of the DV-coordinate. By default is 45;

Top border: upper limit of the DV-coordinate. By default is 55;

Number of bins: `<nbins>`, number of bins to construct the reference data. For cycle 14A the recommended value is 100;

The window also contains two check boxes: **Induced** and **Automatic**.

Induced: selection of this option means that the reference data will be constructed for a gene product stained in a different channel (\neq `<channel>`) of the microscope. This option is only possible regarding the data that are previously registered using expression patterns of other gene stained in the same set of embryos;

Automatic: automated feature extraction by the selected method (`<method>`). If the box is not checked the program operates in interactive mode. In this mode the automatically found features can be corrected manually using GUI (see 6.6).

The reference pattern, if available, can be viewed by pressing the **Edit** button. The pattern is presented on the graph together with the template GCPs (if available). The GCPs' positions can be corrected interactively.

Pressing the **Copy** button allows to use the predefined templates and reference patterns. The set of the ready-to-use reference patterns and GCP templates are provided for 9 segmentation genes *knirps* (*kni*), *giant* (*gt*), *even-skipped* (*eve*), *fushi tarazu* (*ftz*), *hairy* (*h*), *runt* (*run*), *odd-skipped* (*odd*), *paired* (*prd*) and *sloppy-paired* (*slp*) in WT embryos at cycle 14A. This cycle is subdivided into 8 time intervals about 6.5 minutes long each, and the templates are separately computed for each time interval from 2 to 8.

FRDWT window

Define parameters controlling FRDWT feature extraction method. The window is presented in Fig. 7

Xmin: lower limit of GCP coordinate. The left border of the area where GCP are located. If this value is taken too big the first peak may be missed in a pattern where the first peak is located strongly shifted to anterior compared to the reference one. If it is set too little some extra peaks can appear in the anterior area of the spline and the location of the first peak itself can be found with a bit poorer precision. The extra peaks are not critical as they will be rejected by the algorithm. E.g., for *eve* the recommended value is 20-25.

Xmax: upper limit of GCP coordinate. The right border of the area where GCP are located. If this value is taken too little the last peak may be missed in a pattern where the last peak is located strongly shifted to posterior compared to the template value. If it is set too big some extra peaks can appear in the posterior area of the spline. The latter variant is not critical as these extra peaks will be rejected by the algorithm. E.g., for *eve* the recommended value is 90-100;

Delta: minimal amplitude of a first peak. This parameter is used to reject minor peaks of no interest in the anterior area. If it is taken too little an extra peak which may occur in the anterior area of the spline may be mistaken for a first peak, otherwise if it is taken too big the first peak may be missed. The recommended value is a half of a minimal height of the first peak in the dataset.

Spline window

Define parameters controlling Spline feature extraction method. The window is presented in Fig. 8

Xmin: lower limit of GCP coordinate. See the FRDWT window;

Xmax: upper limit of GCP coordinate. See the FRDWT window;

Delta: minimal amplitude of a first peak. See the FRDWT window;

Number of knots: maximal number of knots of a quadratic spline. This value is recommended to be equal to the number of GCPs plus 2-4. The actual number of knots is found by the method automatically and the extra knots are later rejected. This parameter may influence the quality of the spline approximation and may be varied by a user. However, it is hard to predict what is the optimal value of the parameter,

so in the most complicated cases it may be selected experimentally. This parameter is changed in very rare cases when the approximation provides a very poor fit to the data, i.e., the most of pattern extrema don't coincide with the extrema of the spline.

Precision: precision of the Powell optimization method. The recommended value is 0.01.

Smoothing: number of runs of the smoothing algorithm. It is used for smoothing the expression pattern to make it possible to select initial values of knots. The bigger is this parameter the smoother is the curve, hence the more minor extrema of no interest are eliminated and the easier is to find extrema location, and hence initial values of knots. However, if this parameter is taken too big the actual peaks may be eliminated and hence some of necessary knots may be missed. The recommended value for sharp patterns is 20–30, for smooth ones 10–20. By default is set 20.

Figures window

Define the format of visual representation of feature extraction and registration results. The external program gnuplot is used to plot graphs. The input parameter value can be any that gnuplot understands with 'set term' command (fig, tif, png, etc.). If the gnuplot 'term' parameter is 'X11' then the graphs will be displayed immediately in the terminal mode and not saved. Otherwise the graphical files are saved.

6 Program Usage

The feature extraction and registration methods are implemented from the **Actions** window (see Fig. 9).

6.1 Construction of preliminary GCP template and reference pattern

Option **Build template**. Creates the preliminary reference pattern against which individual patterns will be registered. The template GCPs are extracted from the reference pattern. Pressing the **Apply** button opens the **Selected members of dataset** window shown in Fig. 10. The left panel **Dataset** contains the list of files from input directories defined in the **Data** window. The user selects a file and presses the **Add** button to move it to the **Selected** panel. Pressing the **Remove** button discards the selection and moves the file back to the **Dataset** panel. Any number of files can be selected. The selected patterns are grouped into the <Number of bins> bins and averaged to construct the preliminary reference pattern. The individual patterns are plotted on the **Individuals** panel and the resultant reference pattern is presented on the **Averaged** panel.

Template GCPs are selected interactively on the reference pattern using GUI (see 6.6) and saved into the file `int.<tclass>`. The reference pattern is saved into the file `s.<gcode>.<tclass>` if `<method>=spline` or `w.<gcode>.<tclass>` if `<method>=frdwt`.

6.2 Registration of dataset and Update of template and reference data

The option **Register dataset** or **Build/Update reference data** is only activated if the box **Induced** hasn't been checked in the **Reference** window. Registers all the members of the dataset and optionally updates the reference data and template GCPs. Pressing the

Apply button opens the dialog box **Update reference data**. Clicking 'Yes' activates the creation or update of GCP template and reference data. The data files are processed one by one. The program checks whether `<basename>.gcp` files exist for all the data files in the output directory and otherwise runs the feature extraction procedure.

GCPs are extracted automatically by the selected `<method>`. If the checkbox **Automatic** is not checked in the **Reference** window the program is run in interactive mode and the extracted GCPs can be edited manually in the **Verify GCP position** window (Fig. 11). Results of feature extraction are saved in `<basename>.spl` or `<basename>.frdwt` files. The selected (and corrected) GCPs are written into `<basename>.gcp` file. The registered data are saved in `<basename>.reg` and `<basename>.rg` files. Transformation parameters are written into `<basename>.txt` file. If the gnuplot parameter is set X11 the graphs are displayed on screen otherwise the graphical files are saved. The reference pattern is saved in the file `s<tclass>.<gcode>.<nbins> if <method>=spline` or `w<tclass>.<gcode>.<nbins> if <method>=frdwt`.

The results of registration can be displayed in the **Registered data** window (Fig. 12). The left **All** panel contains the output directory defined in the **Data** window and containing the registered data. The user selects a file and presses the **Add** button to move it to the **Selected** panel. Pressing the **Remove** button discards the selection and moves the file back to the **All** panel. Any number of files can be selected. The selected patterns after registration are plotted on the **After registration** panel and the same patterns before registration are presented on the **Before registration** panel.

The **Other** boxes in the lower part of the window allow to input parameters to display the data from other data channels, i.e., the result of Induced registration. The parameter **channel** is the number of the channel with data to be displayed; **One letter code** L, lowercase one-letter identifier of the gene to be displayed. The selected patterns are displayed on the **After registration** and **Before registration** panels after pressing the **Apply** button.

The button **Information** in the **Actions** window opens the **Registered data** window that allows to visualize the registration results. This option only visualizes the previously registered data.

6.3 Registration of an individual pattern

Option **Register pattern**. Registers an individual pattern against the reference one. Pressing the **Apply** button opens the **Select pattern to register** window. The window shows the Open File Dialog to select a data file. The user can create and save shortcuts to directories for the fast access to the data.

GCPs are extracted automatically by the selected `<method>`. If the checkbox **Automatic** is not checked in the **Reference** window the program is run in interactive mode and the extracted GCPs can be edited using GUI (see 6.6) in the **Verify GCP position** window.

Results of feature extraction are saved in the `<basename>.spl` or `<basename>.frdwt` files. The selected (and corrected) GCPs are written into the `<basename>.gcp` file. The registered data are saved in the `<basename>.reg` and `<basename>.rg` files. Transformation parameters are written into the `<basename>.txt` file. If the gnuplot parameter is set X11 the graphs are displayed on screen otherwise the graphical files are saved.

The button **Information** in the **Actions** window opens the **Registered data** window that allows to visualize the registration results. This option only visualizes the previously registered data.

6.4 Update of induced reference data

Option `Build/Update induced reference data` is only activated if the box `Induced` has been checked in the `Reference` window. Builds/updates the reference patterns of genes that have been registered using the pattern of another gene stained in the same set of individuals. This option requires the preliminary registration of the dataset.

The new reference pattern is constructed by grouping and averaging all the registered patterns of the gene. The updated reference pattern is displayed on the `After registration` panel. The reference pattern is saved in the file `s<tclass>.<gcode>.<nbins>` if `<method>=spline` or `w<tclass>.<gcode>.<nbins>` if `<method>=frdwt`.

6.5 Output files

The files created by `Build template`, `Register dataset` or `Build/Update reference data` and `Build/Update induced reference data`.

`int.<tclass>` contains the GCP template. The file is saved in the `output_ref` directory.

`s<tclass>.<gcode>.<nbins>` files containing reference patterns of the gene coded by `<gcode>`. The GCPs are extracted by the Spline method. The file is saved in the `output_ref` directory.

`w<tclass>.<gcode>.<nbins>` files containing reference patterns of the gene coded by `<gcode>`. The GCPs are extracted by the FRDWT method. The file is saved in the `output_ref` directory.

The list of files created by the program for each data file: `<basename>.<filter>`.

`<basename>.reg` The registered 2D pattern (x -coordinates are transformed). The header contains information from BREReA enclosed in tags: `<GCPReg>` and `<\GCPReg>`. Version string of the code enclosed in tags `<version>` and `<\version>` together with modification date; transformation parameters enclosed in tags: `<affine registration>` and `<\affine registration>`. They are: order (`<order>`, `<\order>`), `coefficient_0`, ... , `coefficient_<1>` (`<coefficient_i>`, `<\coefficient_i>`). The file is saved in the `output_tclass` directory;

`<basename>.rg` Contains 1D data from 10% strip after registration. Used to check registration quality in graphical mode. The graph is plotted in superposition with the reference pattern saved in `int.<tc>` file. The file is saved in the `output_dir` directory;

`<basename>.txt` Contains affine transformation parameters. The file is saved in the `output_dir` directory;

`<basename>.spl` Contains spline approximation and 1D data from 10% strip before registration. Used to check approximation quality in graphical mode. The file is saved in the `output_dir` directory;

`<basename>.frdwt` Contains wavelet decomposition and 1D data from 10% strip before registration. Used to check the accuracy of wavelet decomposition in graphical mode. The file is saved in the `output_dir` directory;

`<basename>.gcp` Contains the GCPs extracted by either Spline or Wavelet method. Used to check the extracted features. The file is saved in the `output_dir` directory;

`<basename>.coef` file of the spline coefficients. The work file used by the Feature Extraction procedure.

Graphical files created by the program for each data file: `<basename>.<filter>` if the gnuplot parameter is set different to X11. The files are saved into the `output_dir` directory. For example, if the `xfig` format is defined then the files are:

`<basename>.spl.fig` The spline graph superimposed on the graph of expression pattern. The data are taken from the `<basename>.spl` and the central 10% strip of `<basename>.<filter>` files.

`<basename>.frdwt.fig` The graphs of FRDWT low and high pass superimposed on the graph of expression pattern. The data are taken from the `<basename>.frdwt` and the central 10% strip of `<basename>.<filter>` files.

`<basename>.gcp.fig` The GCPs marked on the graph of 1D expression pattern. The data are taken from the `<basename>.gcp` and the central 10% strip of `<basename>.<filter>` files.

`<basename>.fig` The registered expression pattern superimposed on the reference pattern. The data are taken from the `int.<tc>` and `<basename>.rg` files.

Note: An advanced user can create his own GCP template on the basis of the thorough knowledge of the data and feature extraction methods. The file `int.<tc>` is created manually in the given format (see 8.2.1) and saved in the `output_ref` directory. The user can also optionally provide a reference pattern in the `s<tc>.<gcode>.<nbins>` or `w<tc>.<gcode>.<nbins>` file and save in the same directory. The file format is given in 8.2.2.

6.6 Interactive Graphic Interface

The Graphical User Interface is implemented to correct the automatically detected GCPs. It allows to locate the gcps by clicking the mouse on the graph and to save them into the file.

The code is written in Java for graphics, it communicates with core processing libraries via JNI (C shared library) and is started from 'BREReA' script by running the shell script 'jreg'.

It reads the `<embryo>.gcp` file produced by the previous steps of BREReA and displays the pattern (10% strip) and detected GCPs as colored dots on it with labels (Fig.14). The user then can click the right button at any point on the graph and the popup menu appears. It gives the coordinates of the cursor and the list of actions. If the cursor is in the vicinity of a detected GCP, the action 'Remove' is included into the list. Choosing of this item removes an existing GCP. Otherwise the list includes only the types of GCPs such as 'X1', 'X2', etc. If the user chooses one of these items the corresponding GCP is located and marked out on the graph. If the user wants to shift an existing GCP to a correct location he may just locate it in a new place without previous removing of the old one. The old GCP will disappear automatically.

After all the correct GCPs are marked out the user should press the ‘Save’ button on the lower panel of the graphic interface, then the window is closed, GCPs are saved to the file and the program continues - registers the embryo. It is unnecessary to mark all the extrema, but only those which you consider to have a well-defined position.

7 Program Implementation: Brief User guide

In this section a very brief user guide is given. The more detailed presentation of the method is described in the rest of the document.

Create profile A new profile is created to register a set of expression patterns of a given gene at a given time of development. To create the profile define all the data directories and for each directory define the filter (extension) of data files and the number of the microscope channel in which the gene is stained. All the parameters in the **Reference**, **Spline** (or **FRDWT**) windows should be input. To choose the correct parameter values follow the instructions given above. Note! To run the program in automatic mode (the checkbox ‘Automatic’ in the ‘Reference’ window) it is necessary to get sure that all the patterns are well-resolved with well-localized extremal features (e.g. late *eve* patterns).

Build template Apply the ‘Build template’ option in the ‘Actions’ window. Using the GUI choose one or several the most typical expression patterns with the well-resolved extremal features, average them and select the template GCPs on the averaged pattern. First, it is recommended to mark all the extremal features and then remove those that will be not used for registration. For better registration results it is recommended to choose as GCPs those characteristic features the position of which have the least variance at the given developmental time. The choice should be based on the thorough inspection of all the individual patterns in the dataset.

Register Dataset To register a set of expression patterns in either automatic or interactive mode choose the ‘Register Dataset’ option in the ‘Actions’ window.

Register Pattern This option is applied if there is a need to only register one pattern, for example, to re-register the incorrectly registered one.

The **Registration** is implemented using the following procedures:

Spline Approximation and Feature Extraction is implemented if `<method>` is ‘spline’. The procedure uses the parameters `Xmin`, `Xmax`, `Delta`, `Number of knots`, `Precision` and `Smoothing`. The result of Spline Approximation may be viewed on a graph. The graph of 1D pattern is plotted together with its spline approximation (see Fig.13). The accuracy of fit is checked visually. If the fit is not satisfactory, i.e., the extrema of the pattern do not coincide with the extrema of the spline, the parameters may be changed in the ‘Spline’ window, following the instructions (5.2) and the program is rerun once more with the new parameters. See also Error diagnostics 8.3.3.

The Feature extraction module produces the set of GCP to be used in Registration. The results are visualized by a graph. To check the correctness of selection of GCP the graph of the pattern is plotted with the GCP marked out. See Fig. 14. Note that those GCP are only shown that are further used in registration.

If the GCP are classified incorrectly, e.g., an extremum is misclassified, or less than 3 features are detected, there are two options: to adjust the parameters or/and to rerun the program in interactive mode ('Automatic' is not checked in the 'Reference' window). In this mode

Wavelet Decomposition and Feature Extraction – is implemented if <method> is 'frdwt'. The procedure uses the parameters Xmin, Xmax, Delta. The result of Wavelet decomposition may be viewed mode on a graph. The graph of 1D pattern is plotted together with the low and high passes (see Fig. 13). The accuracy of fit and the correspondence of zeros of high pass to the pattern extrema is checked visually. If the fit is not satisfactory, i.e., the extrema of the pattern do not coincide with the zeros of the high pass, the parameters may be changed in the 'FRDWT' window, following the instructions (5.2) and the program rerun once more with the new parameters.

The Feature extraction module produces the set of GCP to be used in Registration. The results are visualized by a graph in the same way as it is done for the Spline Method.

Registration – the results of Registration are presented as a registered pattern output into the file. Registration coefficients a and b are computed by the least squares method and saved in the separate file <basename>.txt.

The output file is <basename>.reg and it's structure is the same as the structure of the source file, with the x coordinates of nuclei substituted by their affine transformation $ax + b$. The information about registration is put into the header.

The separate file <basename>.rg with the 10% strip is output as well.

To check the quality of registration the 1D graph of the registered pattern is superimposed on the graph of the reference pattern I_i , $i = 1, \dots, 100$ and either displayed or saved in the graphical file <basename>.*, see Fig.15.

The results of registration of several patterns can be viewed in the 'Register Dataset' window.

The **Update of GCP template and reference data** includes:

Processing of Dataset. For each file from the data input directories <data_dir> it is checked whether there is a file with previously computed GCPs in the output directory <output_dir>. If the file <basename>.gcp doesn't exist the program runs the feature extraction procedure in either automatic or interactive mode. The computed GCPs are saved in the files <basename>.gcp.

Update of GCP template. All the GCP which are read from the files <basename>.gcp for each individual embryo are averaged and the registration template is constructed. The updated template is saved in the file int.<tclass>. The file contains the averaged GCP which will be used for the registration of newly obtained embryos belonging to the given <tclass>.

Update of reference data. All the registered patterns of the given gene in the dataset are grouped into the <Number of bins> bins and averaged to construct a new reference pattern. The updated reference pattern is saved in the file that replaces the old file s.<gcode>.<tclass> or w.<gcode>.<tclass>.

The recommended strategy:

Select the most typical patterns and construct the preliminary template and reference pattern.

In the case of well-resolved pattern select the automatic mode:

- register the bunch of files using the ‘Register Dataset’ option automatically. The results of registration are saved as the text result files `<basename>.reg`. The graphical presentation of the results is either displayed on screen or saved in graphical files.
- The graphs visualizing the accuracy of registration must be inspected visually by the user and those patterns for which the registration fails (or is of unsatisfactory quality) are selected.
- Select the interactive mode and rerun the registration for each of these patterns using the ‘Register pattern’ option. Correct the erroneous GCPs manually using the GUI. The other possibility is to correct the program parameters and rerun all the bunch of files automatically once more.

In the case of poorly-resolved pattern select the interactive mode:

- register the bunch of files using the ‘Register Dataset’ option interactively. For each pattern the results of feature extraction are presented in the interactive window and the erroneous GCPs are corrected manually. The results of registration are saved as the text result files `<basename>.reg`. The graphical presentation of the results is either displayed on screen or saved in graphical files.

Update the GCP template and the reference data and re-register the data against the new template.

8 Program Implementation: Detailed Description

Below the detailed presentation of the methods is given.

8.1 Input of the data

The data are input from the directories defined in the **Data** window. The files are processed one by one. The file name is composed of `<basename>.<filter>`. The data records are extracted from the strip defined by **Bottom border** and **Top border** defined in the **Reference** window (see 5.2). A number of the channel containing the expression pattern to be registered is a parameter `channel` input in the **Data** window.

Structure of a data file:

```
*****
# comment lines
...
nucleus # x-coord y-coord 1 channel data [optional] 2 channel data [optional] 3 channel data
*****
```

Example of a data file:

```
*****
# comment lines
```

```

# comment lines
0  0.854948  42.537400  82.001731  0.000000  15.068856
1  1.172000  44.804700  77.090513  0.000000  13.092463
2  1.759100  47.222200  108.009691  0.000000  14.592068
3  2.020420  39.294100  114.663652  0.000000  21.335432
...

```

```
*****
```

The data are read off into the following arrays:

x_i - x -coordinate of a nucleus
 C_i - *eve* expression in a nucleus
 $i = 1, \dots, N$

We read off all the data for which the y -coordinate is between `Bottom border` and `Top border`.

8.2 Creation of preliminary template and reference pattern

Aim: To construct a set of standard GCPs that are used for the registration of individual patterns. A new template can be created in three ways. The first method is recommended in case of expression patterns with well resolved features. Then the user can just select a most typical pattern and use the coordinates of its extrema as GCPs. The second mode is mostly applicable if there is a need to register poorly resolved patterns, from which it is not possible to extract well-defined GCPs. In this situation the user selects a set of individual patterns and applies BREReA to average these patterns and extract GCPs from the averaged pattern. Finally the user can explicitly define the GCP template and save this information as a file in the `<output_ref>` directory. The choice of characteristic features should be based on the thorough analysis of expression patterns of a given gene at a given time interval in the whole dataset. For better registration results it is recommended to choose as GCPs those characteristic features, which positions are the most stabilized at the given developmental time. The number of GCPs is an input parameter of the program. A typical or averaged pattern used to extract GCPs is called reference pattern and saved for visualization of registration results into the `s.jgcode;.itclass;` or `w.jgcode;.itclass;` file. In case of user-defined GCPs the corresponding reference pattern may be optionally provided by the user and saved in the `<output_ref>` directory.

The template GCPs are defined manually using the interactive graphical user interface (see). The user defines source directories containing unregistered data and selects either one or several files each containing quantitative data on expression of a given gene at a given time interval. If several files were selected, the input patterns are automatically averaged and the resultant reference pattern is presented on the lower panel. To extract GCPs from the reference pattern the user places the mouse pointer on each extremum of the pattern diagram from left to right. This opens the pop-up menu, displaying the coordinates of the selected GCP and the list of the feasible GCPs. Click by the right mouse button on the highlighted GCP number writes automatically the x -coordinate of this point into the `int.<tclass>` file. The procedure is repeated until all the points will be extracted. If the expression pattern is not well-resolved and some of the peaks are not formed in all the patterns and/or their position is not yet stabilized, the user may exclude them from the template set. To exclude a GCP the user points it on the graph and selects the **Remove** option in the pop-up menu. Then this GCP is marked by zero in the `int.<tclass>` file.

The files `s.<gcode>.<tclass>`, `w.<gcode>.<tclass>` and `int.<tclass>` are saved in the `<output_ref>` directory.

8.2.1 Input of the GCP template

Input of x coordinates of the template GCPs from the first line of the file `int.<tclass>`. The second line contains units/zeros at the places of features which are used/not used in registration. For the not well-resolved patterns those minima/maxima which are not formed yet in the most of embryos are not used in registration and they are marked by zeros in the second line. We will refer to the set of units as a registration mask. The file `int.<tclass>` is either created by the `Build template` procedure, or copied from the set of ready-to-use templates provided by the program, or directly created by the user.

Example: file `int.3` created for the registration of *eve* pattern from time class 3:

```
*****
32.3 38.9 42.7 45.1 49.4 55.3 57.9 62.8 66.5 70.8 75.1 78.4 83.6
  1   1   1   0   0   1   0   1   0   0   0   1   1
*****
```

The input arrays are

Av_i - coordinates of average extrema

$mask_i$ - unit/zero

$i = 1, \dots, \langle nGCP \rangle$

Av_i is a maximum if i is even and a minimum if i is odd.

8.2.2 Input of the reference pattern

The reference pattern is read off the file `s<tclass>.<gcode>.<nbins>` or `w<tclass>.<gcode>.<nbins>`, containing the reference pattern. This file is either created by the `Build template` procedure, or copied from the set of ready-to-use templates provided by the program, or directly created by the user. The structure of the file is the following:

number; x coordinate of an averaged nucleus; 1D reference pattern; standard deviation

Example of the file `s<tclass>.e.100`

```
*****
  0   0.50  0.44  3.52
  1   1.50  0.73  4.21
  2   2.50  0.73  4.77
  3   3.50  0.49  4.69
  ....
*****
```

The reference pattern is read into the array

I_i ,

$i = 1, \dots, \langle nbins \rangle$

8.3 Spline Approximation

Aim: This module approximates the one-dimensional expression pattern by a quadratic spline. The spline method is a procedure for approximating a function piecewise by polynomials of a given order which are constrained to follow certain continuity requirements

at the borders (“knots”) of the approximated segments. This method provides a smooth approximation of the curves and enables us to estimate extrema in the presence of noise. In a quadratic spline, the function and its derivative are continuous across a knot. Is implemented if `<method>` is set ‘spline’.

8.3.1 Initial values of knots

To find the initial values of knot locations from the experimental expression pattern C_i it is necessary to smoothen the experimental curve to eliminate the noise. This is done by applying the mean filter with the window size $N/(Xmax - Xmin) + 1$. To provide the better smoothing this procedure is repeated several times (input parameter `<Smoothing>`). Then the extrema of the smoothed curve are found and the initial values of knots are taken as middle values between two adjacent extrema. The first and last knot locations are fixed and taken equal to $Xmin$ and $Xmax$, respectively.

8.3.2 Approximation procedure

The definition of the quadratic splines and detailed description of the optimization procedure are given in Appendix.

Given the sample of $\{x_i\}_{i=1}^N$, the x coordinates of N nuclei under consideration, and $\{C_i\}_{i=1}^N$, the values of protein intensity in each nucleus, the $(3 + 2M)$ parameters of this approximation are estimated by the least squares method. Among the parameters M are the coordinates of knots and the other $(3 + M)$ are the parameters of the spline. The optimization is done as a two step procedure, in which the knot locations are found by the Powell optimization method (Press,88) in conjunction with the linear Gaussian method for estimation of other parameters of the spline. Coefficients and an extremum of each parabola composing a spline are represented through these $(3 + 2M)$ parameters. Each parabola is given by the formula $y = ax^2 + bx + c$ or, equivalently, by $y = C0 + k \cdot (x - X0)^2$, where $X0$ is an x coordinate of the extremum, $C0$ is a value of intensity in the point $X0$, k is a coefficient characterizing the slope of a peak.

For each pattern 2 resulting work files are created:

`<basename>.coef` - file of the spline coefficients, which is used in Feature Extraction.

`<basename>.spl` - text file with the approximation data to provide the possibility to view the accuracy of fit by the Spline Approximation.

Structure of the result files:

`<basename>.coef` – file of the spline coefficients. Is used by the Feature Extraction module.

Example:

File structure :

Number of knots.

knots.

a b c : parabola coefficients $y = a * x^2 + b * x + c$.

x0 C0 k : parabola coefficients $y = C0 + k * (x - X0)^2$, where $X0$ is x-coord and $C0$ is intensity in an extremum

```

15                (M - number of knots)
24.32            (knot)
4529.90   -349.14   6.76   (a b c)
25.82      21.65    676.00 (X0 C0 k)

30.03            (knot)
-14388.66  910.46  -14.20 (a b c)
32.04      198.87 -1420.63 (X0 C0 k)

```

```

...
*****

```

<basename>.spl – text file with the approximation data. The file is created and displayed as a graph in terminal mode. The file contains four columns: nucleus number; x coordinate of the nucleus; intensity in the nucleus; spline approximation in the nucleus. An example of a graph see in the Fig.13.

Example:

```

*****
nucl  x-coord    intensity  spline_approx
0     22.295200  25.171700  30.540385
1     22.912400  28.000000  27.439094
2     23.003700  25.800000  27.200899
3     23.246000  27.263700  26.844373
...
*****

```

8.3.3 Error diagnostic

If the program terminates by an error so that the approximation fails, the message indicating the type of an error is output. Instructions: if an error diagnostic is

“Too many steps” - usually it is sufficient to change the parameter # Precision in the file “parameters”.

“Determinant=0” - usually it means that the parameter # Smoothing in the the file “parameters” was set incorrectly.

8.4 Feature Extraction by Spline Method

Aim: This module finds the GCP for registration from the Spline Approximation module. We select the extrema of the pattern which are used as GCP among the extremal points of the spline. Then the valid extrema of the pattern are put into correspondence with the predefined average features (8.2.1), which are the locations of extrema of the reference pattern. Is implemented if <method> is set ‘spline’.

8.4.1 Input of features

We extract features of the pattern from the file <basename>.coef (??) obtained as a result of Spline Approximation.

We read off the data from the file into the following arrays:

$E_i.t_1, E_i.t_2$: two adjacent knots
 $E_i.x$: x-coordinate of the extremum of i th parabola (X0)
 $E_i.C$: intensity in the extremum of i th parabola (C0)
 $E_i.a, E_i.b$ and $E_i.c$: coeffs of spline.
 $i=1, \dots$, number of knots

We introduce one more sequence $E_i.max$ specifying each extremum as either maximum $E_i.max = 1$ or as a minimum $E_i.max = 0$. Note that:

- if $E_i.a > 0$ i th extremum is a minimum, i.e. $E_i.max = 0$
- if $E_i.a < 0$ i th extremum is a maximum, i.e. $E_i.max = 1$
- $E_i.t_1 = E_{i-1}.t_2$

The valid extrema are selected among extremal points of the spline. The spline may contain some extra minor peaks either beyond the pattern area, or on the pattern itself. These extra peaks should not be mistaken for valid peaks of the pattern used as GCP for registration.

At first step the x -coordinate of the first peak is found as a minimal value of $E_i.x$ such that:

- $E_i.max = 1$ (maximum)
- $E_i.x > Xmin$
- $E_i.t_1 < E_i.x < E_i.t_2$ (between two knots)
- $E_i.C \geq Delta$ (minimal height of 1st peak)

The rest of extrema are extracted such that satisfy the following rules:

- $E_i.t_1 < E_i.x < E_i.t_2$ (between two knots)
- $Xmin < E_i.x < Xmax$
- $E_i.max = 1$ if i is even (maximum)
- $E_i.max = 0$ if i is odd (minimum)
- $|E_i.C - E_{i-1}.y| > Amplitude$ (minimal amplitude)

$i = 2, \dots, N$

8.4.2 Standardization of the extracted features

At this step we distinguish between late (having 7 well-defined stripes) and early patterns. A pattern is classified as late if it is from not earlier than 4th time class ($tclass \geq 4$) and number of valid extrema, $N \geq 13$. For late patterns the 13 valid extrema are taken as 13 GCP of the pattern:

$$GCP_i = E_i.x$$

$$i = 1, \dots, 13$$

For early patterns the algorithm is more complicated as some of peaks (stripes) may be missing (or merged) and we must specify to what minima/maxima the extracted valid extrema (E_i) correspond and to reject the extra ones.

8.4.3 Standard features.

To find what extrema we have it is necessary to put their coordinates into correspondence with standard coordinates : $St_i, i = 1, \dots, \langle nGCP \rangle$ These standard coordinates are computed from the reference pattern which is shifted and resized according to the following recursive rules:

$$St_1 = E_1.x$$

$$St_i = St_{i-1} + (Av_i - Av_{i-1}) \cdot (50 - E_1.x)/(50 - Av_1)$$

so that the first standard coordinate coincides with the coordinate of the first extracted maximum; and the difference between two adjacent standard features is equal to the difference between two corresponding average features multiplied by the resizing factor. If the program is run in a semiautomatic mode and some of (or all) the standard coordinates are defined by a user then the corresponding values of St_i are substituted by the predefined ones.

8.4.4 Ground control points

The GCP are found by standardizing the extracted features against the set of standard features. We define for each extracted minimum/maximum the number of the closest standard minimum/maximum as

$$k_1 = 1$$

$$\text{if } E_i.max = 1 \text{ (maximum) } k_i = \arg \min_{j:\{j \text{ even}; j \geq k_{i-1}\}} (|E_i.x - St_j|)$$

$$\text{if } E_i.max = 0 \text{ (minimum) } k_i = \arg \min_{j:\{j \text{ odd}; j \geq k_{i-1}\}} (|E_i.x - St_j|)$$

$$\text{if } k_i = k_{i-1} \quad k_i \text{ is not defined.}$$

$$i = 2, \dots, N$$

The last case may occur if two extracted minima/maxima are the most close to the same standard minimum/maximum, and hence one of them is not considered as a valid GCP.

NB: This point is the most vulnerable in the method. Due to the variability of the shape, location of stripes, and distances between them in the early patterns it happens rather often that some of extrema of the source pattern are classified incorrectly which causes the incorrect registration. In this case it is necessary to input additional information about location and order number of some of (or all) the extrema, which will help to avoid such errors. This possibility is provided in the semiauto mode.

The GCP are defined as

$$GCP_j = E_i.x \quad \text{if } \exists i : j = k_i$$

$$GCP_j \text{ is not defined otherwise}$$

$$i = 1, \dots, 13$$

As N for early patterns is usually < 13 some of $j = k_i$ will be not defined, and if $j = k_i$ is not defined the i th GCP is omitted and not used in registration.

8.4.5 Output results

In terminal mode the results of Feature Extraction are visualized by a graph. To check the correctness of selection of GCP the graph of the pattern is displayed with the GCP marked out. See Fig. 14. The GCP are saved in the text file $\langle \text{basename} \rangle .gcp$

Structure of the file:

Number of GCP

1 line: '*' stands over the GCPs that are used in registration (correspond to zeros in the registration mask).

2 line: standard features

3 line: x-coordinates of GCP

4 line: intensity values at the GCP

Example:

13

*	*	*	*	*	*			*	*		*	*
30.42	36.31	41.64	46.73	50.54	54.98	0.00	0.00	63.93	68.90	0.00	77.06	82.74
29.51	35.70	41.31	46.66	50.67	55.33	0.00	0.00	64.74	69.97	0.00	78.55	84.52
144.46	70.78	138.73	58.32	113.88	42.95	0.00	0.00	62.36	52.82	0.00	33.29	84.66

8.5 Wavelet decomposition and Feature Extraction

Aim: The fast redundant dyadic wavelet transform (FRDWT) is applied to decompose a signal representing 1D *eve* pattern into two sequences: “high pass” and “low pass”. The high pass contains the information on the features of the input signal, while the characteristics of the first derivative are put into the high pass. The zeros of high pass represent a good estimation of the extrema of the original signal. Is implemented if `<method>` is set ‘frdwt’.

8.5.1 Fast redundant dyadic wavelet transform

The detailed description of the FRDWT is given the Appendix 2.

The basic properties of the FRDWT are noise reduction and good localization of characteristic features. It decomposes the signal into two sequences: “high pass” and “low pass”. The number of observations in each sequence equals that in the original sample because of the redundancy of the transform and hence no information is lost about localization of the extrema. At each level of the decomposition the input signal is smoothed by removing the noise of a certain frequency and is placed into the low pass. The high pass contains the information on the features of the input signal, which are determined by the choice of the appropriate wavelet basis (Unser,96). To localize the extrema, i.e. zeros of the first derivative, we apply basis functions that put the characteristics of the first derivative into the high pass. The decomposition is repeated iteratively at each next step decomposing the low pass instead of the original signal.

8.5.2 Feature extraction: Zeros of high pass

At each step of decomposition the signal is represented by two sequences:

Yh_i – high pass

Yl_i – low pass

$i = 1, \dots, N$

As the high pass is a discrete sequence its zeros are defined as a middle of an interval between two adjacent elements of the high pass sequence having a different sign:

$$E_{i,x} = (Yh_i + Yh_{i+1})/2$$

if $Xmin \leq Z_i \leq Xmax$ and

$$Yh_i \cdot Yh_{i+1} < 0$$

$$i = 1, \dots, N_z$$

Besides the sequence specifying each zero as corresponding to either maximum or minimum of the original signal is defined as

$$E_{i,max} = 1 \text{ (ith zero corresponds to a maximum) if } Yh_i > 0 \text{ and } Yh_{i+1} < 0$$

$$E_{i,max} = 0 \text{ (ith zero corresponds to a minimum) if } Yh_i < 0 \text{ and } Yh_{i+1} > 0$$

$$i = 1, \dots, N_z$$

At the first steps of decomposition the first derivative has many extra zeros due to the presence of high frequency noise. At each further step the low pass is getting more and more smoothed and hence the number of zeros of the high pass decreases. The recursive procedure is repeated until the number of zeros N_z doesn't exceed the predefined maximal number of zeros `Zero_number`. If N_z exceeds `Zero_number` one passes to the next level of decomposition, otherwise exits the decomposition procedure. If $N_z = \text{Zero_number}$ the set of zeros extracted at the final step is explored to reject the extra ones (especially at the edges) and to compose the set of valid features.

The case of $N_z < \text{Zero_number}$ means that the required number of zeros never can be achieved, which happens mostly for the early patterns. In such a situation we return back to the previous level of decomposition where the number of zeros still exceeds the critical number. Then the borders of the GCP area $Xmin$ and $Xmax$ are corrected to find such values of $Xmin'$ and $Xmax'$ that at the next step of decomposition the high pass would have the number of zeros located between $Xmin'$ and $Xmax'$ the most close to `Zero_number`. The corrected borders are sought within the predefined intervals: $Xmin - L \leq Xmin' \leq Xmin + L$ and $Xmax - R \leq Xmax' \leq Xmax + R$.

8.5.3 Standardization of the extracted features

See 8.4.2.

8.5.4 Output of results

For each pattern a resulting work file is created:

`<basename>.frdwt` – text file with the wavelet decomposition data. Is created in Verbose mode (Level3) and displayed as a graph in terminal mode to provide the possibility to view the low and high pass at the level of decomposition used for the extraction of GCP.

Structure of the result file:

The file contains five columns: nucleus number; x coordinate of the nucleus; intensity in the nucleus; low pass value; high pass value. An example of a graph see in the Fig.??.

Example:

```

*****
nucl  x-coord    intensity  low pass  high pass
0     22.295200  25.171700  30.540385 3.56939
1     22.912400  28.000000  27.439094 8.324232
2     23.003700  25.800000  27.200899 9.366990
3     23.246000  27.263700  26.844373 2.45237
...
*****

```

8.6 Registration and Output

Aim: To register a source pattern against the reference one using GCP extracted by Feature Extraction method. Registration is implemented by affine transformation of the x -axis so as to minimize the total distance between the corresponding GCP and average coordinates of the extrema over a time class.

The affine transformation $ax + b$ is found by minimizing the sum of squared differences between average coordinates of extrema and GCP:

$$\sum_{i:\{k_i \text{ is defined; mask}_i \neq 0\}} (Av_i - GCP_i)^2,$$

the sum is taken over the intersection of the set of GCP and the registration mask.

Registration coefficients a and b are computed by the least squares method and saved in the separate file `<basename>.txt`.

The output file is `<basename>.reg` and it's structure is the same as the structure of the source file, with the x coordinates of nuclei substituted by their affine transformation $ax + b$. The information about registration is put into the header.

The separate file `<basename>.rg` with the 10% strip is output as well.

To check the quality of registration the 1D graph of the registered pattern is superimposed on the graph of the reference pattern I_i , $i = 1, \dots, 100$ and saved in the file `<basename>.fig`, see Fig.15.

8.7 Update of GCP template and reference pattern

Aim: to update the template GCPs and the reference pattern. A preliminary GCP template is usually constructed using one or a few representative expression patterns which may not contain the full information about the localization of expression domains of a given gene. Thus it is recommended to update templates from time to time as new data is being accumulated. To update the template and reference pattern the user chooses 'Build/Update Reference data'. Then BREReA checks whether GCP were extracted from all the data files in the directory and `<basename>.gcp` files exist, otherwise runs the feature extraction procedure. The new GCP template is constructed by averaging all the GCP values and the new reference pattern is constructed by averaging all the registered patterns. To build or update the reference data for genes scanned in other microscope channels the 'Build/Update induced reference data' option is used. The user should specify the number of microscope channel with the data and the reference pattern will be built by averaging all the registered dataset.

8.7.1 GCP Averaging

All the GCP which are used in registration (i.e., those for which $mask_j = 1$) are averaged over the dataset and thus new standard features are computed:

$$Av_j = \frac{1}{N} \sum_{all\ dataset} GCP_j; \text{ if } mask_j = 1 \\ = 0 \text{ otherwise.}$$

The standard features are written into the file `int.<tclass>` into the user output directory `<output_dir>`.

Structure of the file:

```
*****
```

```
1 line: average features:  $Av_j$   $j = 1, \dots, 13$ 
```

```
2 line: registration mask:  $mask_j$   $j = 1, \dots, 13$ 
```

```
*****
```

Example:

```
*****
```

```
32.3 38.9 42.7 45.1 49.4 55.3 57.9 62.8 66.5 70.8 75.1 78.4 83.6  
1 1 1 0 0 1 0 1 0 0 0 1 1
```

```
*****
```

8.7.2 Data averaging

To construct the one-dimensional reference data for a given gene the x-coordinates of nuclei in each registered one-dimensional expression pattern are grouped along the A-P axis into `<nbins>` intervals. Then the average fluorescence intensity of a given gene is calculated within each interval over all the embryos from the same time class. The value of `<nbins>` is defined by the requirement to correctly model the averaged nuclear structure of an embryo. A single nucleus is very close to 1% egg length in diameter in cycle 14A and the central part of an embryo, hence R should be taken equal to 100 to model a single row of nuclei.

8.7.3 Induced registration

If gene expression in an embryo is imaged in several microscope channels registration of the data obtained in one channel involves registration of genes scanned in other channels. This kind of registration is referred to as induced registration. If there is a need to register expression patterns of all the genes stained in the embryo it is recommended to choose the expression pattern with the most well-defined features and use it for the direct registration.

8.7.4 Output of results

The updated template is saved in the file that replaces the old template file `int.<tclass>`. The updated reference pattern is saved in the file that replaces the old file `s.<gcode>.<tclass>` or `w.<gcode>.<tclass>`.

Appendix 1

Quadratic splines

The simplest approximation is provided by a quadratic spline with M flexible knots, where first derivatives are constrained to be continuous at each knot. The system of knots, $\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_M$, is introduced as a set of distinct points on the x axis, which border the area of each peak. Each knot is located at a point between a peak and the adjacent valley. For late patterns the number of knots is equal to the total number of minima and maxima on the given curve plus two. Early patterns may require a larger number of knots. The quadratic spline with M knots can be represented as

$$sp_2(x) = \sum_{k=0}^2 \zeta_{k0} x^k + \sum_{n=1}^M \zeta_{n2} (x - \tilde{x}_n)_+^2, \quad (4)$$

where $(x)_+ = \max(x, 0)$.

Optimization procedure.

Given the sample of $\{x_i\}_{i=1}^N$, the x coordinates of N nuclei under consideration, and $\{C_i\}_{i=1}^N$, the values of protein intensity in each nucleus, the $(3+2M)$ parameters of this approximation are to be estimated by the least squares method, minimizing the cost function

$$S_1 = \sum_{i=1}^N \{sp_2(x_i) - C_i\}^2.$$

The partial derivatives of S_1 with respect to the spline parameters are set to zero and for the given set of knots $\{\tilde{x}_j\}_{j=1}^M$ the parameters $\{\zeta_{0k}\}_{k=0}^2$ and $\{\zeta_{n2}\}_{n=1}^M$ are estimated by solving the system of $(3 + M)$ linear equations:

$$\begin{aligned} \sum_{k=0}^2 \zeta_{0k} A_{kj} + \sum_{n=1}^M \zeta_{n2} B_{nj} &= D_j, \quad j = 0, 1, 2; \\ \sum_{k=0}^2 \zeta_{0k} B_{lk} + \sum_{n=1}^M \zeta_{n2} C_{nl} &= G_l, \quad l = 1, \dots, M, \end{aligned}$$

where $A_{kj} = \sum_{i=1}^N x_i^k x_i^j$; $B_{nj} = \sum_{i=1}^N (x_i - \tilde{x}_n)_+^2 x_i^j$; $C_{nl} = \sum_{i=1}^N (x_i - \tilde{x}_n)_+^2 (x_i - \tilde{x}_l)_+^2$; $D_j = \sum_{i=1}^N C_i x_i^j$; $G_l = \sum_{i=1}^N C_i (x_i - \tilde{x}_l)_+^2$.

As can be seen from the equation (4), although the cost function is linear with respect to the spline parameters zetas, it is quadratic with respect to the knot parameters, and so a non-linear optimization procedure for their estimation is required. The overall optimization is done as a two step procedure, in which the knot locations are found by the Powell optimization method (Press,88) in conjunction with the linear method for estimation of other parameters of the spline. The Powell method is a direction set method in multidimensions that produces mutually conjugate directions. The basic idea behind Powell's Method is to break the M dimensional minimization down into M separate 1D minimization problems. Then, for each 1D problem a binary search is implemented to find the local minimum within a given range. Furthermore, on subsequent iterations an estimate is made of the best directions to use for the 1D searches.

At each step of the non-linear procedure the exact minimum of the cost function S_1 is found by solving the linear system for the current set of knots by the Gauss method.

As a result of such approximation each curve is specified by a set of knots $\{\tilde{x}_j\}_{j=1}^M$ and by the spline parameters $\{\zeta_{0k}\}_{k=0}^2$ and $\{\zeta_{n2}\}_{n=1}^M$. From these parameters the x -coordinate of the k -th peak's extremum is given by

$$X_k = \frac{2 \sum_{n=1}^k \zeta_{n2} \tilde{x}_n - \zeta_{01}}{2(\zeta_{02} + \sum_{n=1}^k \zeta_{n2})}, \quad k = 1, \dots, M.$$

Appendix 2

Fast redundant dyadic wavelet transform

The FRDWT is defined by the equation

$$(W_\psi f)(a, k) = a^{-1/2} \int_{-\infty}^{+\infty} \overline{\psi} \left(\frac{x-k}{a} \right) f(x) dx, \quad (5)$$

where $k \in Z$ (the integers), a is the frequency window scaling factor and $\overline{\psi}$ denotes the complex conjugate of ψ , the wavelet. We use the set of functions

$$Q = \left\{ \psi_{j,k}(t) = 2^{-j/2} \psi(2^{-j}t - k); \quad j, k \in Z \right\}$$

that are obtained by translation and dilation of a single wavelet function $\psi(t)$:

$$\begin{aligned} (W_\psi f)(2^j, 2^j k) &= \langle f(t), \psi_{j,k}(t) \rangle \\ &= 2^{-j/2} \int_{-\infty}^{+\infty} f(t) \overline{\psi(2^{-j}t - k)} dt. \end{aligned}$$

Thus FRDWT uses a set of variable window sizes that are proportional to 2^{-j} and maps a function $f(t)$ into two double sequences: low pass $s_j(k) = (W_\phi f)(2^j, 2^j k)$ and high pass $r_j(k) = (W_\psi f)(2^j, 2^j k)$, where ψ is a wavelet and ϕ is a scaling function. The transform operates at all dyadic scales to extract information about the signal. We choose $\psi(t) = \frac{3}{4}t \exp(-t^2/2)$ which extracts information about the first derivative and $\phi(t) = \frac{3}{4} \exp(-t^2/2)$ which is responsible for smoothing.

Discrete implementation.

Even if the wavelet transform is usually expressed in terms of integrals, it can still be computed exactly through an appropriate sequence of discrete convolutions (digital filters). This is only possible because of the general Hilbert space framework, which allows one to represent the input signal f and the wavelet in terms of linear combinations of shifted basis functions, so that

$$\begin{aligned} f(x) &= \sum_{k \in Z} c(k) \phi(x - k), \\ \psi(x) &= \sum_{k \in Z} p(k) \phi(x - k), \end{aligned}$$

and the function $\phi(t) = \frac{3}{4} \exp(-t^2/2)$ (where $\frac{3}{4}$ is the normalizing coefficient).

k	-2	-1	0	1	2
$b(k)$	0	0.167	0.667	0.167	0
$h(k)$	0.125	0.5	0.75	0.5	0.125
$p(k)$	0	-0.5	0.5	0	0

Table 1: The filter coefficients used in computation.

The functions ϕ and ψ represent the desired shape of the scaling function and the wavelet. From computational point of view it is better to use the compact supported functions of the desired shape. Any function from L_2 can be approximated with splines of degree n . B -splines of degree n form an unconditional Riesz basis in space of splines of degree n . B -splines are compactly supported. Consequently, for any discrete signal $f(k)$ there is a unique representation

$$f(x) = \sum_{k \in \mathbf{Z}} f_0(k) \beta^n(x - k), \quad (6)$$

where $f(x)|_{x=k} = f(k)$.

Then

$$f(k) = (b^n * f_0)(k) \leftrightarrow f_0(k) = ((b^n)^{-1} * f)(k), \quad (7)$$

and

$$b^n(k) \equiv \beta^n(x)|_{x=k} \quad (8)$$

is the discrete B -spline kernel of degree n , $(b^n)^{-1}$ denotes the inverse filter operator, which exists and is stable for any degree n .

Mallat and co-workers constructed an algorithm of extraction of features of the first derivative based on cubic B -spline wavelets (Mallat,92). The function $\phi(t) = \beta^3(t)$ has a bell shape and both functions ϕ and ψ have compact support.

Hence, in order to analyze the discrete signal $f(k), k \in I_N = 0, \dots, N-1$, one starts by computing

$$s_0(k) = \sum_{l=-N_b}^{+N_b} b(l) f(k+l)$$

where $b(k), k = -N_b, \dots, N_b$ is the initialization kernel.

Next, we compute the inner products between the input signal and the scaling functions at the various dyadic scales :

$$s_j(k) = \sum_{l=-N_h}^{+N_h} h(l) f(k + 2^{j-1}l), \quad j = 1, \dots, J.$$

where $h(k), k = -N_h, \dots, N_h$ is the refinement filter.

The transform is then computed by convolving these signals with the dilated version of the wavelet

$$r_j(k) = \sum_{l=-N_p}^{+N_p} p(l) s(k + 2^j l), \quad j = 0, \dots, J.$$

where $p(k), k = -N_p, \dots, N_p$ is the wavelet coefficient sequence. For $(k + ml) \notin I_N$ periodic boundary conditions are used: $f(-1) = f(N-1), \dots, f(N) = f(0), \dots$

The explicit values of filter coefficients are found in the Table 1. They are obtained from the representation of functions ϕ and ψ with respect to the properties of the wavelet basis. The detailed procedure can be found in (Unser,96).

References

- E.Myasnikova, M.Samsonova and J.Reinitz. "Removal of Background Signal from in situ Data on the Expression of Segmentation Genes in Drosophila", 2005, *Development, Genes and Evolution*, 215(6):320-326.
- Brown, L. G. (1992). "A survey of image registration techniques." *ACM Computing Surveys*, 24(4):325–376.
- Janssens, H., Kosman, D., Vanario-Alonso, C.E., Jaeger, J., Samsonova, M., and Reinitz, J. (2005). "A high-throughput method for quantifying gene expression data from early Drosophila embryos" *Development Genes and Evolution*, 215:374-381
- S.Mallat and W.L.Hwang : "Singularity detection and processing with wavelets". *IEEE Trans. Inform Theory*, **38(2)**, 1992, 617-643.
- K. Kozlov, A. Samsonova, S. Surkova, J. Reinitz and M. Samsonova "GCPReg package for registration of the segmentation gene expression data in Drosophila", , *Fly* 3(2): 151-156, 2009
- E. Myasnikova, A. Samsonova, K. Kozlov, M. Samsonova, and J. Reinitz. "Registration of the expression patterns of *Drosophila* segmentation genes by two independent methods." *Bioinformatics*, 17:3–12, 2001.
- Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. (1988). *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge.
- M.Unser: "A Practical guide to the implementation of the wavelet transform" in *Wavelets in Medicine and Biology*, A.Aldroudi and M.Unser, eds, CRC Press, 1996, 37-73.
- Surkova, S., Myasnikova E., Janssens H, Kozlov K, Samsonova A, Reinitz J, Samsonova M. "Pipeline for acquisition of quantitative data on segmentation gene expression from confocal images". *FLY* 2008; 2(2): 58-66

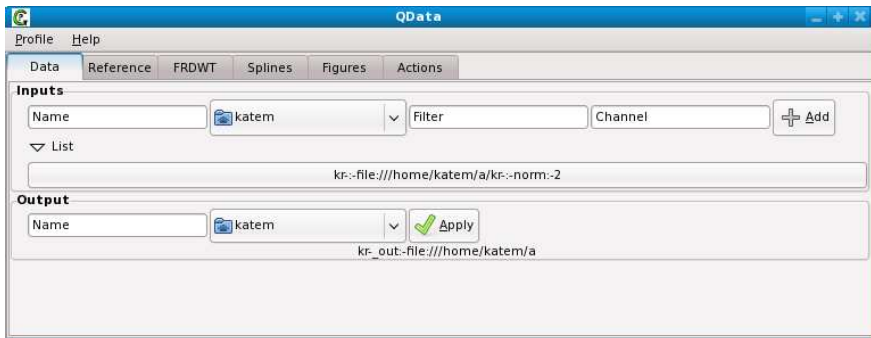


Figure 5: Screenshot of the Data window. Input of input and output directories.



Figure 6: Screenshot of the Reference window. Input of parameters necessary to construct GCP template and reference data.

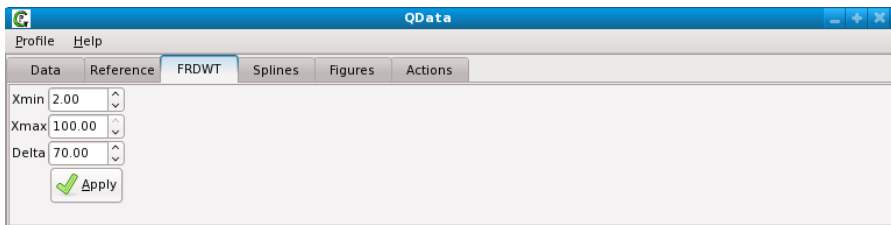


Figure 7: Screenshot of the FRDWT window. Input of parameters controlling FRDWT feature extraction.



Figure 8: Screenshot of the **Spline** window. Input of parameters controlling the Spline feature extraction.

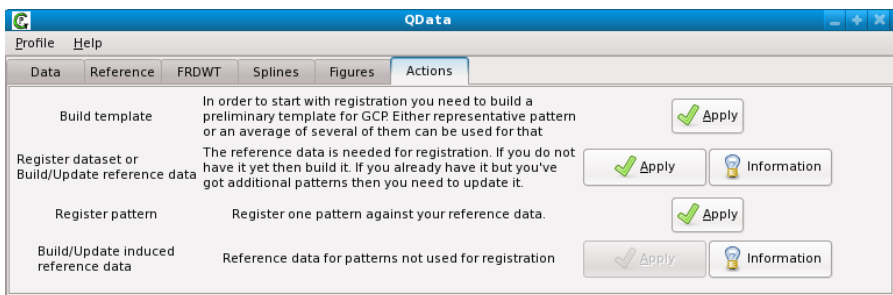


Figure 9: Screenshot of the **Actions** window. BREReA implementation.

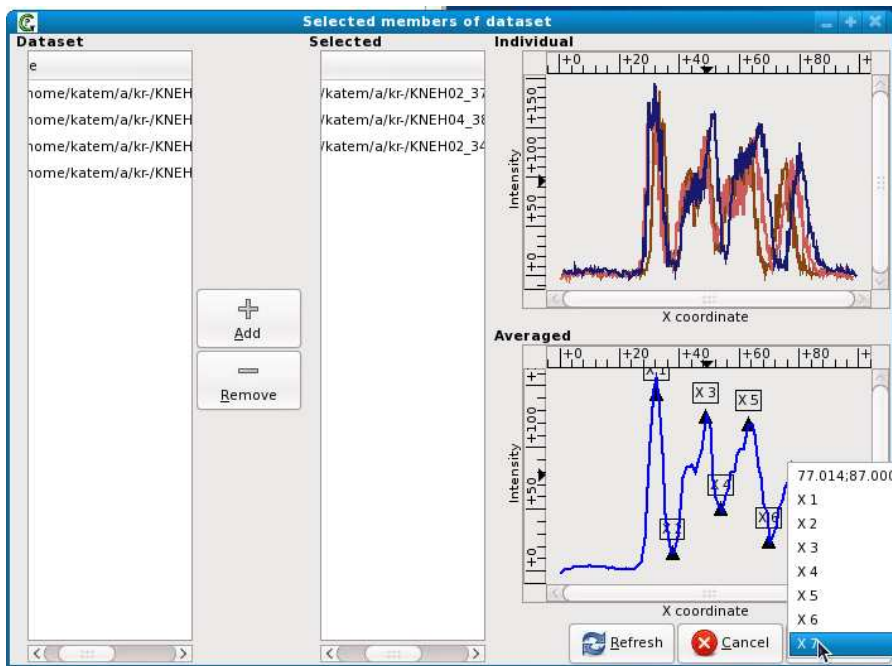


Figure 10: Build/Update template.

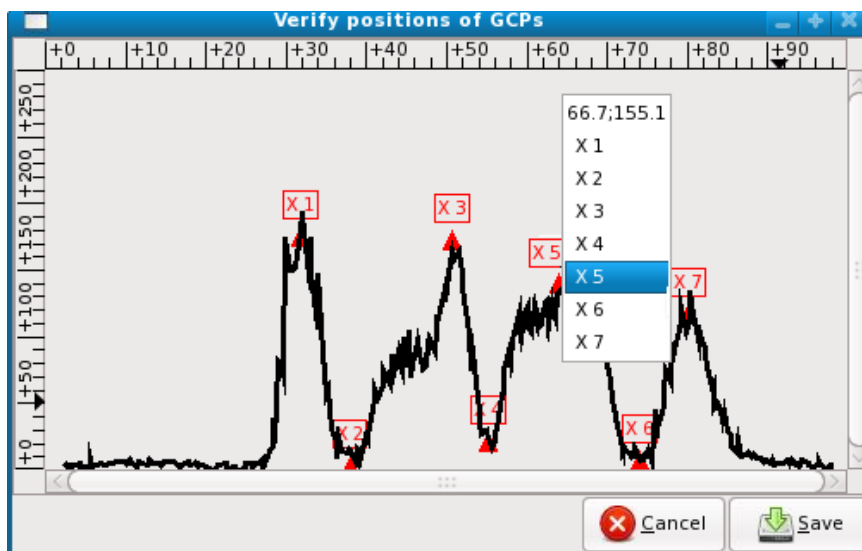


Figure 11: Graphical User Interface to correct the extracted GCPs manually.

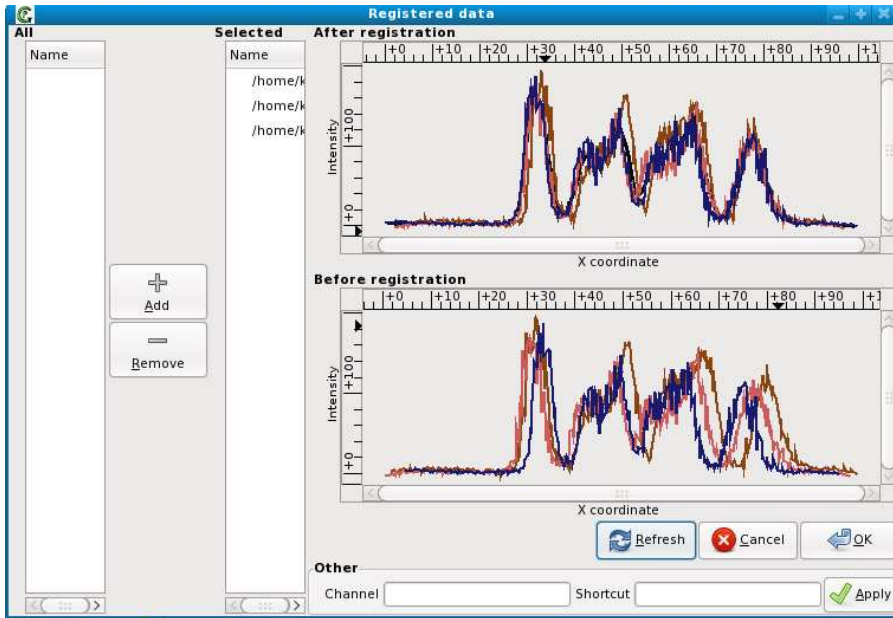


Figure 12: Register Dataset. Build/Update reference data.

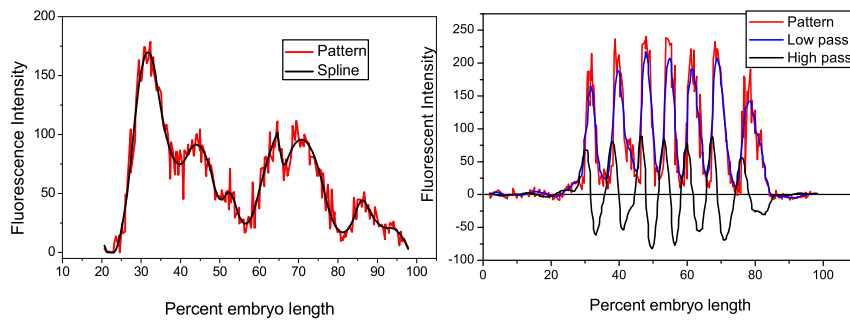


Figure 13: Examples of spline approximation and FRDWT decomposition..

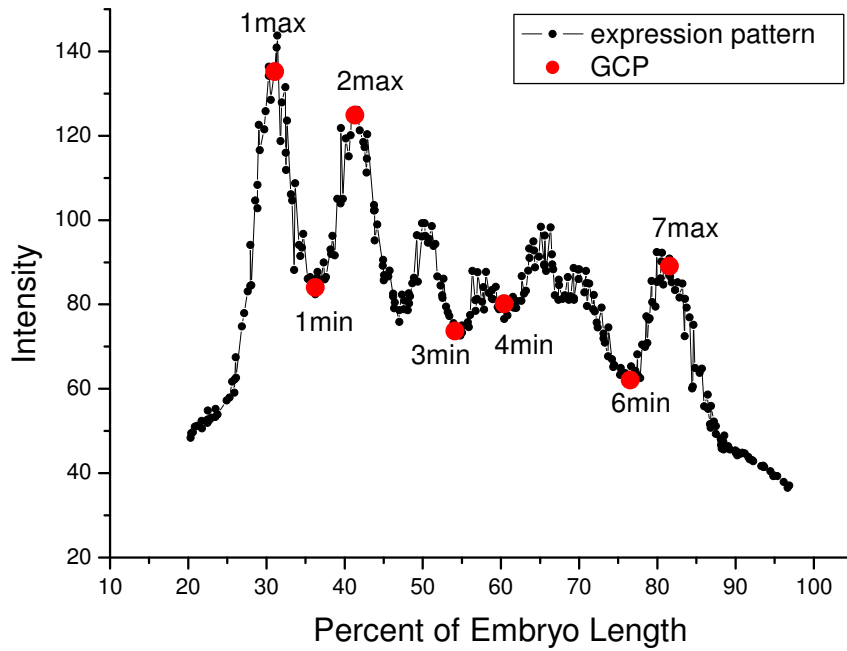


Figure 14: An *eve* expression pattern belonging to T3 with the GCP marked out red. Some of the extrema detected by Spline Approximation or Wavelet Decomposition are not presented as they are not used in registration of T3 embryos.

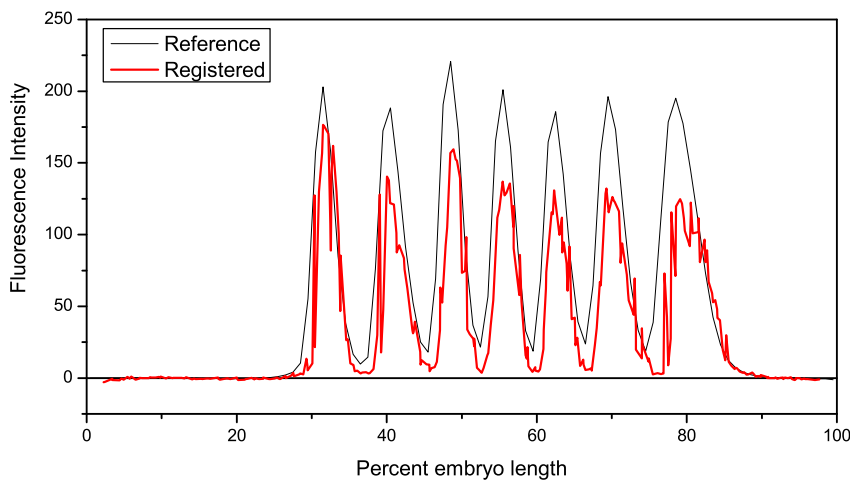


Figure 15: An 1D *eve* expression pattern superimposed on the reference one. The graph allows to check the accuracy of registration.