USER-FRIENDLY IMAGE-BASED SEGMENTATION AND ANALYSIS OF CHROMOSOMES

*V. Uhlmann, R. Delgado-Gonzalo, M. Unser**

Biomedical Imaging Group,
EPFL, Lausanne, Switzerland

P. O. Michel, L. Baldi, F. M. Wurm

Cellular Biotechnology Laboratory,
EPFL, Lausanne, Switzerland.

ABSTRACT

We designed two efficient and user-friendly tools for the segmentation and analysis of images containing chromosomes or, more generally, rod-shaped elements that are spread on microscopic slides. The segmentation tool allows to automatically extract the profile of each chromosome and to sort the collection of profiles in a karyotype image. The analysis tool is interactive and allows to extract quantitative measurements and annotate the relative position of the centromere to the chromosome extremities in a fast and reproducible way. The two methods rely on custom variants of parametric active contours. Both have been designed as user-friendly plug-ins for the open-source software ImageJ.

***Index Terms*—** Active contours, segmentation, chromosomes, image analysis.

1. INTRODUCTION

Karyotyping is an important operation for genetic analyses such as species or gender determination of eukaryotic cells and organisms. It is based on the analysis of the number and morphology of chromosomes that become compacted into rod-like structures during metaphase, the first short phase of the cell division process. More generally, the assessment of chromosomal size as well as the analysis of shape parameters (such as centromere position and sizes of chromosome arms) through the microscopic analysis of metaphase spreads is a widely applied technique for the study of genomic structure, organization, function and evolution in different fields of biological and environmental sciences [1, 2, 3].

Automated computer-based chromosome analysis tools have been developed since the seventies, including early methods based on densitometric scanning of standard photographs [4, 5]. These approaches remained anecdotic due to the poor computational power available at the time. More recently, the availability of multi-purpose open-source image analysis softwares enables biologists to process and annotate chromosome images. However, most of the processing is still performed in a fully-manual fashion, accounting for large amount of operator time as genetic experiments usually involve large collections of data.

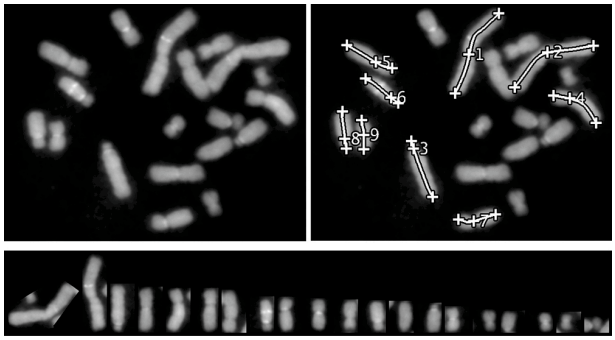

Fig. 1. Chromosome segmentation and analysis pipeline. Raw image (top left), chromosome annotation using the annotation tool (top left), and karyotype image generated from the raw image using the segmentation tool (bottom).

In this paper, we present a relatively simple and cost-effective method based on digital image processing for karyotyping and performing chromosome analysis (Figure 1). We designed two pieces of software that can be used together or separately in order to segment and analyze chromosome images. First, the segmentation tool aims at extracting the image profile of each chromosome and sorting them by size in a karyotype image. The underlying algorithm relies on standard image processing methods along with a custom model of parametric active contours. More precisely, it is composed of four steps: image normalization, detection of candidate regions via data clustering and pruning, outlining with active contours, and extraction of the profile of the chromosomes. Then, the analysis tool allows to precisely extract numerical measurements such as lengths and centromere positions in images of chromosomes. The output is provided in a standard format to allow further data processing. The approach is mostly automated and requires minimum user input. Thus, it reduces the intra- and inter-user variability and speeds up the annotation process.

Both tools have been programmed as plug-ins for ImageJ, which is a free open-source multiplatform Java image-processing software [6]. They do not depend on any special-

*This work was funded by the Swiss National Science Foundation under Grants 200020-144355 and 200020-121763.

ized hardware and, through ImageJ, any common image format may be used. The two plug-ins are freely available online along with their companion user manual[1]. We dedicate this paper to the description of the algorithmic details of the segmentation tool (Section 2) and of the annotation tool (Section 3). Both softwares have already been successfully used for practical applications in biology [7], significantly speeding up the image analysis and data extraction steps.

2. DETECTION AND SEGMENTATION OF THE CHROMOSOMES

In this section, we provide the algorithmic details of the segmentation tool, which extracts the chromosomes and gathers them in a karyotype image containing the different chromosomes sorted by size.

2.1. Image Normalization

To improve the robustness of the segmentation, the image is first locally normalized to zero mean and unit variance over some neighborhood. This process is repeated in sliding fashion over the whole image. This is especially useful to correct for nonuniform illumination or shading artifacts. The estimation of the local mean and standard deviation is performed through local spatial smoothing using fast recursive Gaussian filters.

2.2. Detection of Candidate Regions

Standard k-means clustering [8] is then performed on the normalized image in order to partition all pixels into k classes. The best results, allowing to separate the inner from the outer part of the chromosomes is achieved by clustering the image in $k = 3$ classes. In this situation, the class with the lowest mean is identified as the background, the class with the highest mean contains the pixels that represent the chromosomes, and the remaining class contains the pixels that belong to a transition zone between the background and the chromosomes.

The k-means clustering provides a set of pixels that represents the foreground objects, but the algorithm does not discriminate between different chromosomes within the foreground cluster. To separate them, the 8-connected components of the foreground cluster is labeled using a linear-time algorithm [9]. Each connected component is a chromosome candidate. However, due to the presence of photometric noise and overlap between different chromosomes, it is necessary to discard candidate regions that do not meet certain criteria. To that end, the centroid of each region, its inertia matrix, and its area are computed. Regions smaller than $\tau_a = 50$ square-pixel units are then discarded. After computing the minimal bounding rectangle along the direction given by the inertia matrix of a candidate region, regions associated to bounding rectangles that are longer than some user-specified value τ_l are finally discarded.

[1] Segmentation tool: http://bigwww.epfl.ch/algorithms/chromosomeJ/
Analysis tool: http://bigwww.epfl.ch/algorithms/chromosomeK/

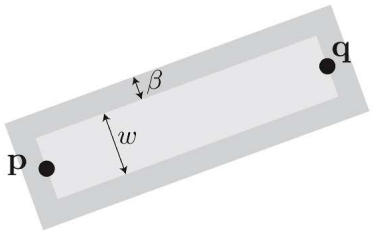

Fig. 2. The rectanguscule. The inner rectangle R_{in}, shown in brighter gray, is of constant width w. The outer rectangle R_{out}, shown in darker gray, has the same centroid and orientation as the inner one and is obtained by extending the boundaries of R_{in} by a distance β. These rectangles are entirely determined by the pair of points $\{\mathbf{p}, \mathbf{q}\}$ that belong to the boundary of the inner one.

2.3. Outlining with Custom Active Contours

Each chromosome is individually segmented using as initialization the bounding rectangles from previous step. For this task, we rely on active contours (a.k.a. snakes [10]). Snakes are effective tools for image segmentation, which consist in a curve that evolves from an initial position towards the boundary of an object. The evolution of the curve is formulated as a minimization problem. The associated cost function is called the snake energy.

To guide the process, the chromosomes are first assumed to have a conserved width. Second, we assume that the shape of the chromosomes can be described by one rectangle or by a composition of at most two rectangles. Finally, the pixels that represent the chromosome are assumed to be brighter than the background.

To efficiently segment bright objects, we use an approach similar to [11] and adapt it to the specifics of the segmentation of quasi-rectangular chromosomes. We name our rectangular snake the *rectanguscule* (Figure 2). The rectanguscule is parameterized by two points $\mathbf{p}$ and $\mathbf{q}$ on the image plane. The inner rectangle, R_{in}, is of constant user-specified width w. The length and orientation are determined by the control points. The area of the inner rectangle is given by $|R_{\text{in}}| = w \, \|\mathbf{p} - \mathbf{q}\|$. The outer rectangle, R_{out}, has the same centroid and orientation as the inner one. It is constructed by extending the boundaries of the inner rectangle by a constant user-specified distance β. The area of the outer rectangle is thus computed as $|R_{\text{out}}| = (2\,\beta + \|\mathbf{p} - \mathbf{q}\|)\ (2\,\beta + w)$.

The corners of the outer rectangle are determined by the

control points according to the relations

$$\mathbf{r}_1^{\text{out}} = \mathbf{q} + \beta\mathbf{u} + \left(\frac{w}{2} + \beta\right)\mathbf{v}, \quad \mathbf{r}_2^{\text{out}} = \mathbf{q} + \beta\mathbf{u} - \left(\tfrac{w}{2} + \beta\right)\mathbf{v},$$
$$\mathbf{r}_3^{\text{out}} = \mathbf{p} - \beta\mathbf{u} - \left(\frac{w}{2} + \beta\right)\mathbf{v}, \quad \mathbf{r}_4^{\text{out}} = \mathbf{p} - \beta\mathbf{u} + \left(\tfrac{w}{2} + \beta\right)\mathbf{v},$$

where $\mathbf{u} = (\mathbf{q} - \mathbf{p}) / \|\mathbf{q} - \mathbf{p}\|$ is the unit vector that determines the orientation of the rectangle and $\mathbf{v}$ is a vector orthonormal to $\mathbf{u}$. Likewise, the corners of the inner rectangle are determined by the control points according to the relations

$$\mathbf{r}_1^{\text{in}} = \mathbf{q} + \frac{w}{2}\mathbf{v}, \qquad \mathbf{r}_2^{\text{in}} = \mathbf{q} - \frac{w}{2}\mathbf{v},$$
$$\mathbf{r}_3^{\text{in}} = \mathbf{p} - \frac{w}{2}\mathbf{v}, \qquad \mathbf{r}_4^{\text{in}} = \mathbf{p} + \frac{w}{2}\mathbf{v}.$$

The rectanguscule is a surface snake whose energy is not driven by the data under the curve, but by the data enclosed by it. At each iteration of the optimization process, the geometry of the rectanguscule is updated to increase the contrast between the intensity of the data averaged over its rectangular core R_{in} and the intensity of the data averaged over its rectangular shell R_{out}. For $R_{\text{in}} \subset R_{\text{out}}$ and the input image data f, the snake energy is defined as

$$E = \int_{R_{\text{out}} \setminus R_{\text{in}}} f(x, y)\, \mathrm{d}x\, \mathrm{d}y - \lambda \int_{R_{\text{in}}} f(x, y)\, \mathrm{d}x\, \mathrm{d}y,$$

where the factor λ is set to

$$\lambda = \left(\frac{|R_{\text{out}}|}{|R_{\text{in}}|} - 1\right).$$

This value of λ enforces that the energy remains zero when f takes a constant value irrespective of the size and orientation of the snake. The position of the control points is tuned using a standard unconstrained optimization algorithm that minimizes the energy of the snake. The final configuration of the control points provides an accurate description of the orientation and size of each chromosome. The optimization of the snake is carried out efficiently by a Powell-like line-search method [12].

2.4. Extraction of the Chromosomes

Finally, we extract each chromosome within the region determined by the rectangular snakes. In some situations, long chromosomes may appear bended with respect to their central section. In these cases, two different rectangular snakes are used to capture each branch of the chromosome. In our software, we allow the user to connect/disconnect rectangular snakes in order to deal with bent chromosomes. The chromosome extraction is performed using cubic-spline interpolation at the resolution of the original image. We sort the extracted chromosomes by the combined length of the rectangular snakes that determine them in order to construct the karyotype image.

3. INTERACTIVE ANALYSIS OF THE CHROMOSOMES

In the following, we describe the design of the analysis tool for interactively extracting quantitative information on the chromosomes.

3.1. Annotation of the Chromosomes

To initiate the annotation process, the user defines the two extremities of a chromosome by mouse-clicking on their location within the image. The two points serve as seeds to automatically draw a curve going through the medial axis of the chromosome. This is obtained by optimizing an open active contour (or open snake). Our formulation differs from classical snakes which are usually defined as closed curves. The open snake is parametrized by $M \geq 2$ control points. The two user-defined extremities delimit the chromosome are considered as fixed anchors. The additional $M - 2$ control points are defined with or without constraints. We define the parametric representation of the curve

$$\mathbf{r}(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \sum_{k=0}^{M-1} \mathbf{c}_k \phi(t - k)$$

with natural cubic spline as basis function $\phi(t)$, as it has been demonstrated [13] that cubic splines are optimal for representing smooth parametric curves of low-curvature. The open snake energy is obtained by integrating the values of the Euclidean-distance-transformed image (f_{EDM}, [14]) under the snake curve, *i.e.*,

$$E = \int_C f_{\text{EDM}}\, \mathrm{d}s = \int_0^1 f_{\text{EDM}}(\mathbf{r}(t))\, |\mathbf{r}'(t)|\, \mathrm{d}t.$$

In order to optimize our open snake, we rely on a multiresolution strategy inspired from [13]. The open snake is first initialized with three control points: the two user-defined extremities, and a point at mid-distance in the segment that joins them. The endpoints are considered as ground truth to define chromosome extremities and are not modified during the optimization process. The unconstrained optimization algorithm described in Section 2.3 is used to modify the free control point position and minimize the energy of the snake. At convergence, the optimal curve is used to spawn a snake offspring composed of five control points, the additional points being set halfway between each pair of parent control points. The three parent points are fixed, leaving only two to be modified. This two-steps approach grants enough flexibility to detect S-shaped chromosomes without loosing precision on U-shaped ones (Figure 3). It ensures that the three first control points are well positioned to match U-shaped objects, and then refines the curve to match objects with more complex geometry. In this way, all chromosome shapes can accurately be segmented in minimal computation time.

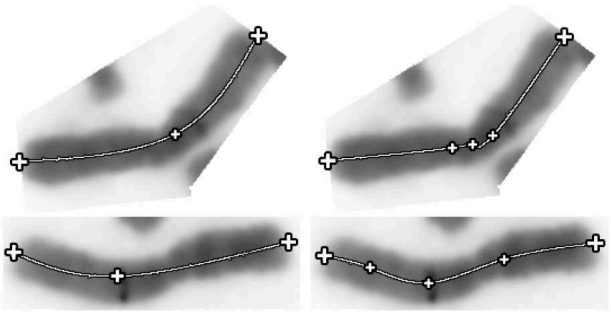

Fig. 3. Variation of the number of control points. U-shaped chromosomes are best annotated using a three control points snake (top left). Five free control points results in irregular curves (top right). On S-shaped chromosomes, a three control points snake doesn't offer enough flexibility to precisely capture chromosome shape (bottom left). When only five control points are used, a more precise trace is obtained (bottom right).

Once the medial axis of a chromosome is detected, the user can define the centromere position by clicking on the trace. A numerical label is assigned to each chromosome based on its length (1 being the longest), with automatic renumbering during image annotation. The curve, centromere, and extremities of any chromosome can be edited at all time. We emphasize that the centromere is only an optional annotation on the curve and is not related in any way to the position of the control points.

3.2. Chromosome Measurements

Chromosome annotations are saved in an XML file that can be reloaded in the software for further edits. The annotated image as well as a comma-separated value file containing helpful measurements such as total chromosome length, centromere position, and relative length of each chromosome arms (if the centromere was defined) are extracted as well. It is worth noting that chromosome traces are recorded in the XML file as a collection of point coordinates. These data can hence handily be used to extract additional shape-based measurements from each object.

4. CONCLUSIONS

Our contribution in this paper are two user-friendly, free, and open-source methods for image-based karyotyping and analysis of chromosomes (Figure 4). The analysis tool can be used independently or as a companion plug-in for the segmentation/karyotyping tool. Although built on rather simple image processing algorithms, our softwares speed up processing time, systematize data extraction, and have already been shown to be useful for biologists. In particular, we point out that while both tools were initially developed for chromosome analysis, they can be used as well for the analysis of other rod-shaped objects, including bacteria, worms, etc.

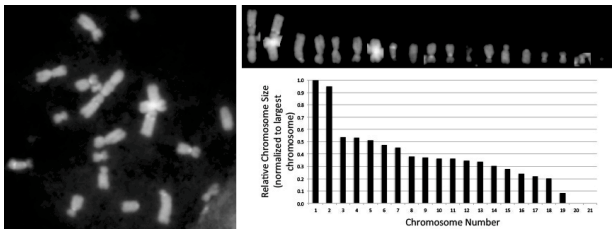

Fig. 4. Example of data analysis. Raw image (left), resulting karyotype (top right), and extracted chromosome lengths (bottom right).

5. REFERENCES

[1] S.A. Romanenko et al., "Karyotype evolution and phylogenetic relationships of hamsters (Cricetidae, Muroidea, Rodentia) inferred from chromosomal painting and banding comparison," *Chromosome Research*, vol. 15, no. 3, pp. 283–298, April 2007.

[2] X. Li et al., "Chromosome size in diploid eukaryotic species centers on the average length with a conserved boundary," *Molecular biology and evolution*, vol. 28, no. 6, pp. 1901–1911, January 2011.

[3] I. Titos, T. Ivanova, and M. Mendoza, "Chromosome length and perinuclear attachment constrain resolution of DNA intertwines," *The Journal of cell biology*, vol. 206, no. 6, pp. 719–733, September 2014.

[4] T.T. Puck, "Action of radiation on mammalian cells III. Relationship between reproductive death and induction of chromosome anomalies by X-irradiation of euploid human cells in vitro," *PNAS*, vol. 44, no. 8, pp. 772, August 1958.

[5] D. Schoëvaërt-Brossault, C. Léonard, and J. Selva, "A new method for automatic metaphase finding adaptable to different chromosome preparations," *Computer programs in biomedicine*, vol. 16, no. 3, pp. 195–201, June 1983.

[6] M.D. Abràmoff, P.J. Magalhães, and S.J. Ram, "Image processing with ImageJ," *Biophotonics International*, vol. 11, no. 7, pp. 36–41, July 2004.

[7] P.O. Michel et al., "Assessment of chromosomal length variation in CHO cells," in *Proceedings of ESACT'13*, Lille, France, June 23-26, 2013, p. 206.

[8] A.K. Jain and R.C. Dubes, *Algorithms for Clustering Data*, Prentice Hall, 1988.

[9] K. Suzuki, I. Horiba, and N. Sugie, "Linear-time connected-component labeling based on sequential local operations," *Computer Vision and Image Understanding*, vol. 89, no. 1, pp. 1–23, January 2003.

[10] R. Delgado-Gonzalo, V. Uhlmann, D. Schmitter, and M. Unser, "Snakes on a plane: A perfect snap for bioimage analysis," *IEEE Signal Processing Magazine*, vol. 32, no. 1, pp. 41–48, January 2015.

[11] P. Thévenaz, R. Delgado-Gonzalo, and M. Unser, "The ovuscule," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 2, pp. 382–393, February 2011.

[12] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, third edition, 1986.

[13] P. Brigger, J. Hoeg, and M. Unser, "B-Spline snakes: A flexible tool for parametric contour detection," *IEEE Transactions on Image Processing*, vol. 9, no. 9, pp. 1484–1496, September 2000.

[14] A. Rosenfeld and J.L. Pfaltz, "Distance functions on digital pictures," *Pattern Recognition*, vol. 1, no. 1, pp. 33–61, July 1968.