SegAnnDB: interactive web-based genomic segmentation
Toby Dylan Hocking 1, Valentina Boeva 2,3,4, Guillem Rigaill 5, Gudrun Schleiermacher 2,6, Isabelle Janoueix-Lerosey 2,6, Olivier Delattre 2,6, Wilfrid Richer 6, Franck Bourdeau 6, Miyuki Suguro 7, Masao Seto 7, Francis Bach 8, Jean-Philippe Vert 2,3,4

1 Tokyo Institute of Technology, Japan, 2 Institut Curie, 26 rue d’Ulm, 75248 Paris Cedex 05, France, 3 INSERM U900, Paris, France, 4 Mines ParisTech, Centre for Computational Biology, 77300 Fontainebleau, France, 5 Unité de Recherche en Génomique Végétale INRA-CNRS-Université d’Evry Val d’Essonne, 6 INSERM U830, Paris, France, 7 Aichi Cancer Center Research Institute, 1-1 Kanokoden, Chikusa-ku, Nagoya-city, 464-8681 Japan, 8 INRIA–Sierra project-team, Département d’Informatique de l’École Normale Supérieure, Paris, France.

ABSTRACT

Motivation: DNA copy number profiles characterize regions of chromosome gains, losses, and breakpoints in tumor genomes. Although many models have been proposed to detect these alterations, it is not clear which model is appropriate before visual inspection the signal, noise, and models for a particular profile.

Results: We propose SegAnnDB, a web-based computer vision system for genomic segmentation: first visually inspect the profiles and manually annotate altered regions, then SegAnnDB determines the precise alteration locations using a mathematical model of the data and annotations. SegAnnDB facilitates collaboration between biologists and bioinformaticians, and uses the UCSC genome browser to visualize copy number alterations alongside known genes.

Availability: The breakpoints project on INRIA GForge hosts the source code, an Amazon Machine Image can be launched, and a demonstration web site is http://bioviz.rocq.inria.fr.

Contact: toby@sg.cs.titech.ac.jp

© The Author(s) 2014. Published by Oxford University Press. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.
selection. Given a particular data set to analyze, it is neither obvious to choose a particular model nor its tuning parameters. How to evaluate which model is best?

Without doing more experiments, the only method of evaluation is to plot the model alongside scatterplots of the data. A good model should capture all visible alterations in the data, and should not predict any false positive detections. This visual criterion for model evaluation can be used by creating a database of annotated regions that encode an expert’s interpretation of the data (Hocking et al., 2013). In that study, default parameter values of several models were shown to yield many false positive and false negative detections. And even after tuning the parameters of each model, there were no models with perfect detection accuracy.

In this article we propose to solve this problem by interactively annotating alterations in scatterplots of the data (Figure 1). Then our software selects a tuning parameter that agrees with the user-defined annotations, and immediately shows the updated model. If the displayed segmentation model does not capture the alterations which are obvious from visual inspection, then annotations can be added to correct the model. A user-specific model can thus be iteratively improved until it agrees with the annotator’s visual interpretation of the data.

In addition, the annotations can be assembled into a database so we can apply algorithms that automatically recognize previously annotated patterns. We note that this computer vision approach has also been successful for recognizing phenotypes in cell microscopy (Jones et al., 2009). After annotating a small subset of the data, the system adapts to the annotations and provides consistent predictions for un-annotated data.

This paper describes SegAnnDB, a web-based free/open-source implementation of this interactive genomic segmentation model. The name is short for Segmentation and Annotation DataBase, since annotations are stored in a database, which is used by machine learning algorithms to find an appropriate segmentation. After interactive annotation, the learned segmentation model can be directly exported to the UCSC genome browser for viewing detected alterations alongside known genes (Kent et al., 2002). Using SegAnnDB facilitates cancer research by (i) allowing biologists to share their interpretations of tumor copy number profiles, (ii) providing a breakpoint and copy number annotation model that agrees with the user’s visual interpretation of the data, and (iii) allowing visualization of several sample profiles at once to spot recurrent patterns.

Finally, SegAnnDB promotes collaboration between biologists, doctors, and bioinformaticians doing genome-wide copy number analysis (Figure 2). Collaboration using SegAnnDB is simple: once a bioinformatician uploads a profile, anyone with a web browser can create a user-specific segmentation model by drawing annotated regions on the scatterplots (Figure 1). This makes it easy for people with expert prior knowledge but no programming experience to browse the profiles and annotate regions of interest (e.g., a biologist looking for alterations in known oncogenes). After annotation, the bioinformatician can download the annotations and segmentation model for further analyses such as detection of common alterations in several related samples, or survival regression based on the detected alterations.

2 SYSTEM AND METHODS

In this section, we describe the general workflow when using a SegAnnDB server for annotation-guided DNA copy number analysis. Later in the Algorithm and Implementation sections, we give details about how the server interactively calculates and displays the models.

2.1 Uploading profiles

The first step of any analysis is to upload the normalized logratio data to the web site. The data should be uploaded in gziped UCSC bedGraph format, since this is the format that will also be used to export the data for viewing on the UCSC genome browser (Kent et al., 2002). The 4-column text-based bedGraph format is simple, so any bioinformatician should be able to quickly convert data from any platform-specific formats.

The bedGraph header line must contain the following three important variables specific to SegAnnDB. The db variable should indicate the genome version of the probe positions (e.g. db=hg19). The maxSegments variable specifies the maximum number of segments per chromosome for the initial SegAnnDB scatterplots, prior to manual annotation (kmax in Section 3.1). The share variable controls who can view the profile.

Fig. 1. General workflow in annotation-guided DNA copy number analysis on SegAnnDB. Scatterplots of black points show logratio as a function of genomic position. Breakpoints in the current segmentation are shown with vertical dashed lines, and predicted copy number status of each horizontal line segment is indicated by its color (Table S2). Annotated regions can be added to update the copy number (top) and breakpoints (bottom) in the displayed model. Dragging on an un-annotated region creates a new region with Save and Delete buttons, as shown for the normal region in the center. Before saving, the annotation can be changed by clicking the region. After saving, the displayed model is immediately updated to agree with the annotation.
SegAnnDB: interactive web-based genomic segmentation

- *share=public* means all users of the website.
- *share=private* means only the user who uploaded it.

Profiles can be uploaded one at a time using the upload-profiles.py command line program. Once a profile has been uploaded, it must undergo two main processing steps before interactive annotation is possible. First, the logratio values and several maximum likelihood segmentation models are saved in BerkeleyDB, a highly efficient database system. Second, PNG scatterplots are generated for several zoom levels (Table S1). The only limit on the number of profiles is the amount of disk space available on the SegAnnDB server. Importantly, the amount of time and disk space required is linear in the profile size (Table 1).

### 2.2 Plotting data and annotating breakpoints

Once a profile has been uploaded and processed, it can be plotted and annotated. From the home page or the list of profiles, clicking a profile name shows a zoomed out plot of all its chromosomes. Plots can be zoomed to individual chromosomes and then zoomed further by clicking the plot size links as shown in Figure 1.

Each plot initially shows the uploaded data as black points and a segmentation model as horizontal line segments (Figure 1). We use vertical dashed lines to draw the “breakpoints,” which are the change-points in the piecewise constant segmentation model.

The breakpoints in the displayed segmentation can be edited by adding breakpoint annotations to the bottom half of the plot. There are two types of breakpoint annotations: 1breakpoint means there is exactly one breakpoint in the region, and 0breakpoints means there are no breakpoints in the region. Dragging on an un-annotated region creates a new region with Save and Delete buttons, as shown in Figure 1. Before saving, an annotation can be changed from 1breakpoints to 0breakpoints by clicking the region. After saving the annotation, it is sent to the server, which calculates the consistent segmentation model defined in Section 3.1. The server immediately sends the consistent segmentation back for display in the web browser. Thus the segmentation model can be iteratively updated by adding breakpoint annotations, until the displayed breakpoint locations match the expert interpretation of the annotator.

As explained in Section 3.1, one of two possible segmentation models will be used: pruned dynamic programming (PrunedDP) or SegAnnot. The color of the dashed vertical breakpoint lines indicates the algorithm: green for PrunedDP and purple for SegAnnot. It is important to note that when the SegAnnot algorithm is used, there must be a 1breakpoint annotation for each breakpoint in the segmentation. In contrast, the PrunedDP algorithm can detect breakpoints in un-annotated regions.

<table>
<thead>
<tr>
<th>Profile size probes</th>
<th>time seconds</th>
<th>BerkeleyDB megabytes</th>
<th>scatterplots megabytes</th>
<th>probes.gz megabytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2,678</td>
<td>1</td>
<td>&lt; 1</td>
<td>1</td>
<td>&lt; 1</td>
</tr>
<tr>
<td>56,855</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>262,230</td>
<td>10</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>1,868,857</td>
<td>100</td>
<td>22</td>
<td>27</td>
<td>28</td>
</tr>
</tbody>
</table>

Table 1. System requirements for profiles of different sizes. The processing time includes calculating segmentation models and PNG scatterplots on a 2.9GHz Intel X7520M CPU. We show disk space occupied by a profile in the database (BerkeleyDB), the total size of the PNG images (scatterplots), and the size of the data file to upload and export to UCSC (probes.gz).

### 2.3 Annotating copy number

After finding a model with appropriate breakpoints, copy number annotations can be added to the top half of each plot to define copy number status (Figure 1). Each copy number annotation should define a region of equal copy number: deletion, loss, normal, gain, or amplification (Table S2). More types of copy number annotations can be defined for specific projects by editing the SegAnnDB source code. After adding or deleting a copy number annotation, SegAnnDB updates the predicted copy number status of each segment of the profile. This is indicated by immediately updating the color of the displayed horizontal line segments. This immediate genome-wide visual feedback is crucial for avoiding accidental mistakes in annotation.

There are two special colors for horizontal line segments: green and black. If a profile has no copy number annotations, then all segments are colored green. If a segment has several different overlapping copy number annotations, it is colored black to indicate that it should be corrected.

Copy number status is generalized across segments of a profile, so not all segments need to be labeled. The segments with overlapping copy number annotations are used to infer the copy number status of the other unlabeled segments. However, one copy number annotation of each type must be present in order to use this feature. For example, if there is only 1 normal and 1 gain annotation on a profile, then the rest of the segments will be labeled as either normal or gain, and there will be no segments labeled loss.

### 2.4 Exporting data to the UCSC genome browser

After a profile has been appropriately annotated, a table of detected alterations can be displayed by clicking the “Show alterations on the UCSC genome browser” link shown on the bottom of Figure 1. Clicking a button on that page exports the probes, annotations, and segmentation to the UCSC genome browser (Figure 2). Although positions of genes can not be viewed while annotating profiles on SegAnnDB, they can be viewed alongside the exported segmentation model on the UCSC genome browser (Figure 3). Data from multiple profiles can be displayed at the same time on UCSC, for rapid visual verification of repeated alterations in particular genes.

There are two methods for quickly navigating from SegAnnDB to altered regions on the UCSC genome browser. First, shifting-clicking any annotated region opens a new web page with the UCSC genome browser zoomed to that region. Second, each profile has an alterations table which contains links to each detected breakpoint, gain, loss, amplification, and deletion.

The annotations and displayed segmentation models can also be downloaded for other analyses. For example, the annotations could be used to develop better models for breakpoint detection or copy number calling. Also, the displayed segmentation model could be used for further analyses such as survival regression or plotting a genome alteration print (Popova et al., 2009).

![Fig. 2. SegAnnDB exports data to the UCSC genome browser and facilitates collaboration between bioinformaticians and biologists. First, profile probe logratio values are uploaded to a SegAnnDB server by an uploader (section 2.1). Then an annotator can plot the data and refine a segmentation model by adding annotations (sections 2.2, 2.3). Finally, an analyst can send data to the UCSC genome browser, which displays plots of the segmentation model with known genes (section 2.4).](http://bioinformatics.oxfordjournals.org/)
3 ALGORITHM

In this section we explain the algorithmic details of the models used to detect breakpoints and copy number alterations.

3.1 Calculating the displayed segmentation model

To find an appropriate segmentation model for each chromosome, we first calculate a sequence of segmentation models, then use the expert’s breakpoint annotations to choose a consistent model.

Let $y \in \mathbb{R}^d$ be the normalized log-ratios for one chromosome, measured at positions $p_1 < \cdots < p_d$. This signal is drawn using black points in Figure 1. Because of its speed and breakpoint detection accuracy (Hocking et al., 2013), we segment using the pruned dynamic programming (DP) algorithm of Rigaill (2010). It calculates the least squares segmentation $\hat{y}^k \in \mathbb{R}^d$ for every $k \in \{1, \ldots, k_{\text{max}}\}$ segments:

$$\hat{y}^k = \arg \min_{\mu \in \mathbb{R}^d} \sum_{i} \| y - \mu \|^2$$

such that $\mu$ has $k - 1$ changes.

The PrunedDP algorithm returns a sequence of progressively more complex segmentation models $\hat{y}^1, \ldots, \hat{y}^{k_{\text{max}}}$. For each model size $k$, the segmentation $\hat{y}^k$ has the least squared error among all models with $k$ segments.

The maximum number of segments $k_{\text{max}}$ is the only parameter of PrunedDP. For interactive annotation on the SegAnnDB web site, we usually set $k_{\text{max}} = 20$, which means that up to 20 segments will be shown on the initial scatterplot, prior to manual annotation. After annotation, if there are 20 or more breakpoint annotations, then SegAnn will be used instead of PrunedDP, as explained below. For high-density arrays with many expected breakpoints (e.g. chromothripsis), larger values may be specified (e.g. $k_{\text{max}} = 200$). However, $k_{\text{max}} = 20$ was a reasonable choice for even the high-density arrays that we analyzed (Figure S1, Table S3).

The model selection problem is to choose one of the $k_{\text{max}}$ segmentation models, which we do using breakpoint annotations. Let $R_0, R_1$ be the sets of 0breakpoints and 1breakpoint annotations, respectively. These appear on the bottom half of the scatterplots shown on SegAnnDB (Figure 1). Each $r \in R_0, R_1$ is an interval of base pairs, so to compare with the segmentation model we need to convert the model breakpoint locations to base pairs. For each model size $k$, we estimate the set of base pairs after which a change occurs using

$$B(\hat{y}^k, p) = \{(p_j, p_{j+1})/2 \} \text{if } \hat{y}^k_j \neq \hat{y}^k_{j+1}, \forall j \in \{1, \ldots, d - 1\}.$$ 

Equation 2 defines a breakpoint for every change $\hat{y}^k_j \neq \hat{y}^k_{j+1}$ in the signal, at the base pair halfway between the probes $(p_j + p_{j+1})/2)$. These breakpoint positions are shown using dashed vertical lines on SegAnnDB scatterplots (Figure 1).

We calculate the annotation error to determine which PrunedDP models agree with the set of current breakpoint annotations. The annotation error $e : \{1, \ldots, k_{\text{max}}\} \to \mathbb{Z}_+$ uses the zero-one loss to judge agreement of the predicted breakpoints $B(\hat{y}^k, p)$ and the annotated regions $R_0, R_1$:

$$e(k) = \sum_{b=0}^{1} \sum_{r \in R_b} I(\{B(\hat{y}^k, p) \cap r \neq b\}).$$

The indicator function $I$ is 0 when the model $\hat{y}^k$ predicts the correct number of breakpoints in a region $r$, and 1 otherwise. If the model with $k$ segments has no annotation error $e(k) = 0$, then we say that the segmentation $\hat{y}^k$ is consistent with the given annotations. The set of consistent PrunedDP models is $K_0 = \{k : e(k) = 0\}$.

If there are any consistent PrunedDP models then $|K_0| > 0$ and we define the optimal number of segments as

$$k^* = \arg \min_{k \in K_0} |k - \hat{k}(y)|,$$

where the predicted number of segments $\hat{k}(y) \in \{1, \ldots, k_{\text{max}}\}$ is learned using the log-sigmoid max margin interval regression model on the other annotated chromosomes (Rigaill et al., 2013). In short, $k$ is a prediction function which minimizes the average annotation error over all annotated chromosomes. So as annotations are added, the predicted set of breakpoints tends to get more accurate (Figure 4). To support fast model updates during interactive annotation, the prediction function $\hat{k}$ is learned in the background on the web server. Equation 4 is used to pick the consistent model $k \in K_0$ which is closest to the complexity of the predicted model $\hat{k}(y)$.

If there are no consistent PrunedDP models then $|K_0| = 0$ and we use the SegAnn dynamic programming algorithm (Hocking and Rigaill, 2012). SegAnn exactly recovers the least squares segmentation such that the annotation error is 0, meaning that each $R_0$ region has 0 breakpoints and

Fig. 3. The data, annotated regions, and labeled segments are exported for visualization alongside RefSeq genes on the UCSC genome browser. From top to bottom, the probe log-ratios are shown in black, the annotated regions are shown using their colors on SegAnnDB, the log-ratio of the segmentation model is shown using a white < black scale, the breakpoints in the segmentation model are red, and the inferred copy number status of each segment is shown using the copy number annotation colors of Table S2. In this example, it is clear that the MYCN oncogene is amplified in this neuroblastoma tumor sample.
each $R_i$ region has exactly one breakpoint:

$$\text{SegAnn}(y, p, R_i) = \arg \min_{\mu \in \mathbb{R}^d} \{ |y - \mu|_2^2 \}
\text{ such that } \forall r \in R_i, |B(\mu, p) \cap r| = 1,$$

$$\text{and } \mu \text{ has no other changes.} \tag{5}$$

The constraint in (5) means that the estimated mean $\mu$ changes once in each breakpoint region $r \in R_i$, and has no changes elsewhere.

In summary, the displayed segmentation is given by

$$\begin{cases}
g^c \leftarrow y & \text{if } |K_0| > 0 \\
\text{SegAnn}(y, p, R_i) & \text{otherwise.} \tag{6}
\end{cases}$$

By construction, this segmentation is consistent with the breakpoint annotations $R_0, R_1$.

3.2 Calculating copy number state

Segments with overlapping copy number annotations are used to infer the copy number status of un-annotated segments on the same profile. Copy number status is inferred by learning a set of up to 4 thresholds, between the 5 possible copy number annotations (Table S2). Each of these thresholds is learned in the same way, which we explain below for just the threshold between normal and gain.

Let $N, G$ be the sets of normal and gain segment means, respectively. Let there be $n = |N|$ normal segments and $g = |G|$ gain segments, and let $\bar{Y}_1, \cdots, \bar{Y}_{n+g}$ be the ordered segment means. We consider thresholds $t_i = (\bar{Y}_i + \bar{Y}_{i+1})/2 \in \mathbb{R}$ for all $i \in \{1, \cdots, n + g - 1\}$. Given a threshold and a segment mean $y \in \mathbb{R}$, we predict copy number status

$$\begin{cases}
\text{normal} & \text{if } y < t_i \\
\text{gain} & \text{if } y > t_i. \tag{7}
\end{cases}$$

The best threshold minimizes the number of incorrectly predicted copy number annotations:

$$\arg \min_{t_i \in \{1, \cdots, n + g - 1\}} \sum_{y \in N} I(y > t_i) + \sum_{y \in G} I(y < t_i). \tag{8}$$

The indicator functions $I$ are 0 when the threshold $t_i$ correctly predicts the copy number status of an annotated segment, and 1 otherwise. The best threshold (8) can be calculated in linear $O(n + g)$ time.

4 IMPLEMENTATION

In this section we discuss the technologies required to implement a web-based interactive system like SegAnnDB.

4.1 Interactive scatterplots

Scatterplots of probe logratio values are displayed quickly on SegAnnDB since they are pre-rendered and saved as bitmap PNG images when a profile is uploaded. Some zoom levels are scaled in proportion to the number of probes, so high-density profiles can result in very large PNG images (Table S1). Some web browsers do not render these large PNG images. For example, current iPad web browsers do not render images wider than 20,000 pixels, so we defined an "iPad" zoom level specifically to accommodate this maximum possible zoom level. Among desktop web browsers, we found that Google Chrome works best on several platforms.

Interactive model updates are implemented using the D3 Javascript library (Bostock et al., 2011). The basic idea is to draw the segmentation model and breakpoints for each chromosome on an HTML web page with an SVG element that has the PNG scatterplot as the background. After each change to the annotated regions, the server updates the displayed model by sending a JavaScript Object Notation (JSON) file to the annotator’s web browser.

4.2 Data storage, export, and server configuration

In order to support interactive annotation and model updates, SegAnnDB needs to store data in a fast database system. We used Berkeley DB since it allows quick storage and retrieval of any data types. In addition, the disk space requirements are approximately linear in profile size (Table 1).

Data can be exported to the UCSC genome browser by clicking a button on the SegAnnDB web site. This sends the copy number data, annotations, and model directly from SegAnnDB to the UCSC genome browser. Figure 2 explains the relationship between the uploader, annotator, SegAnnDB server, and UCSC genome browser.

Configuring the SegAnnDB software on a Debian/Ubuntu GNU/Linux system requires Python and several free/open-source extension packages. For example, the Pyramidal web framework and the SegAnn and PrimedID segmentation modules are required. To avoid the time-consuming task of configuring all these packages for every new SegAnnDB server, we made a public Amazon Machine Image (AMI) so launching a new SegAnnDB server takes only 10 minutes on Amazon's Elastic Compute Cloud. A list of current AMIs can be found in the SegAnnDB source code README on INRIA GForge.

5 RESULTS AND DISCUSSION

5.1 Supervised versus unsupervised analysis

In the machine learning literature, a learning problem is “supervised” when there is a teacher or oracle that provides correct predictions for training an algorithm. In this article, the type of supervision that we propose is a set of annotated regions from an expert’s visual interpretation of the data. These annotated regions are then used by SegAnnDB to build a model with consistent breakpoint locations and copy number calls.

In contrast, most statistical models for DNA copy number analysis can be viewed as unsupervised since they do not use an annotation database for model training. Typically, an unsupervised model is trained on the data, and then an expert plots the model to judge if it fits the data well. This sequence of steps is inverted on SegAnnDB: first, we plot and annotate the data, and then we fit a model to the combined data and annotations.

We considered the DNAcopy circular binary segmentation model of Venkatraman and Olshen (2007) as an unsupervised baseline model, and compared its performance to SegAnnDB. We analyzed the data with DNAcopy, an unsupervised segmentation algorithm, and compared the results to those obtained with the supervised approach displayed in SegAnnDB. The supervised model provides more accurate and consistent results, especially in regions with complex copy number changes, which are challenging for unsupervised methods. This highlights the advantage of combining machine learning with expert knowledge in the interactive annotation process.
speed, train error, and test error with respect to 7 annotation data sets, consisting of 708 neuroblastoma, lymphoma, and medulloblastoma copy number profiles (Table S4). The data come from cell lines and primary tumors (Figure S2) analyzed using different platforms (BAC/PAC, Nimblegen, Affymetrix), so the number of probes per profile ranges from 1719 to 1,686,857 (Table S3). In total, there were 4467 annotated chromosomes containing 6937 annotated regions.

Table S5 shows that the PrunedDP algorithm used by SegAnnDB took under 2 hours to train on the entire data set, but DNACopy took over 3 hours for each of the 31 parameter values we tested. PrunedDP was faster overall because it was faster for high-density profiles (Figure S3).

The displayed segmentation on SegAnnDB is always consistent with the given breakpoint annotations, so SegAnnDB has 0% training error by definition (Section 3.1). In contrast, it may be impossible to achieve 0 annotation errors with an unsupervised model that does not directly use the breakpoint annotations (Hocking and Rigail, 2012). Table S6 shows that the unsupervised dnaCopy.default algorithm predicts 4–50% incorrect breakpoint annotations across the 7 data sets we examined. For a more balanced comparison, Hocking et al. (2013) showed that DNAcopy breakpoint detection can be improved by picking an undo-SD parameter which minimizes breakpoint annotation error (we did not apply smoothing before segmentation, and we kept default values for all other parameters). For each data set we picked chromosome-specific undo-SD values, yielding better training error rates (Table S6, dnaCopy.sd, 0.33–2.75%). Although breakpoints detected by these methods are qualitatively similar (Figure S4), SegAnnDB is quantitatively more accurate with 0% error.

We also evaluated the test error by training a model on a set of annotated chromosomes, and counting the number of incorrectly predicted breakpoint annotations on a test set. Figure 4 and Figure S5 show that adding more annotations improves the test error of the supervised SegAnnDB and dnaCopy.sd methods (data-set-specific parameters were chosen by minimizing annotation error for each randomly chosen training set). Predictably, the supervised methods adapt to the provided annotations and detect breakpoints more accurately than the unsupervised dnaCopy.default algorithm. However, on some high-density data sets, even the best methods showed up to 20–40% test error (Figure S5). The large test error of these supervised methods motivates spending time on interactive annotation, which always achieves 0% training error.

SegAnnDB is a computer vision system that exploits the strong points of the human visual system and mathematical models. The human visual system is good at detecting noise and breakpoints in the copy number signal over large regions, but bad at detecting the precise location of a breakpoint. Mathematical models are good at detecting the precise locations of alterations, if tuning parameters are adjusted appropriately. SegAnnDB combines the best of both approaches by automatically adjusting the tuning parameters of a mathematical model to agree with an expert’s visual annotations.

5.2 Time required for visual annotation on SegAnnDB

The amount of time is proportional the number of annotated regions that need to be added to correct the displayed model. More breakpoint annotations are sometimes necessary for larger profiles (Figure S6), but when SegAnnDB provides good predictions in un-annotated regions (Figure 4), annotating all breakpoints is not necessary (Figure S1). So in our experience it takes about 10 minutes to completely annotate even high-density profiles, and it should be feasible to annotate each profile in data sets of a few dozen samples.

For larger data sets, it may not be feasible to annotate every profile. For example, visually annotating every profile from the 1000 Genomes Project of Altshuler et al. (2010) would require about 10 minutes per profile × 1000 profiles / 60 minutes per hour = 166 hours, which is indeed several weeks of work. But for large data sets SegAnnDB can still be useful for creating a relatively small database of 5 to 10 visually annotated profiles that are representative of the bigger data set (generated on the same platform and for the same tumor type). Then the protocols of Hocking et al. (2013) can be used for training and validating an algorithm for automatically detecting copy number alterations on the un-annotated profiles. The main idea is that for each manually annotated region, one can check if an algorithm predicts the indicated copy number or breakpoints in that region. If there is some disagreement, then the algorithm needs to be improved. So when there is not enough time to visually annotate every profile, an annotated region database is still useful for checking the validity of an automatic annotation algorithm.

5.3 Correcting experimental artifacts

Supervised copy number analysis is useful for correcting the various types of noise and artifacts that can be present in DNA copy number profiles. In this section, we discuss three types of corrections that can be easily visually annotated using SegAnnDB.

First, some profiles have noise patterns that should be ignored, and are easy to visually identify. Examples are wave patterns and outliers, as shown in the left panel of Figure 5. If one of these

![Fig. 5](https://example.com/figure5.png)

**Fig. 5.** Examples of noise and alterations that can be annotated using SegAnnDB. **Left:** outliers and wave noise can be ignored using breakpoints annotations. **Center:** normal copy number at a non-zero log ratio value can be indicated using normal annotations. **Right:** small alterations can be annotated using breakpoints and copy number annotations.
patterns can be visually identified, then SegAnnDB can be used to exclude it by simply placing a breakpoints region around it.

Second, many copy number calling algorithms assume that the baseline normal level of 2 copies should be centered around logratio 0, but this can depend on the normalization procedures. For example, the center panel of Figure 5 shows a profile whose normal level appears at logratio $\approx 0.5$. When it is visually obvious that normal copy number is at a non-zero logratio value, then normal copy number regions can be used to indicate that.

Finally, many algorithms have a parameter for the minimal number of probes required for identifying a copy number alteration. For example, by default DNAcopy sets its min.width parameter to 2 probes. Instead of defining an arbitrary value that may not hold for all alterations in a data set, breakpoint annotations can be used to define small alterations that are visually obvious, such as the loss shown in the right panel of Figure 5. Then, for example, the annotations can be used to check which values of the DNAcopy min.width parameter are appropriate for a given data set.

5.4 Using annotations in later analyses

The annotation database, segmentation, and copy number calls created on SegAnnDB can be used as inputs to other analyses. SegAnnDB creates a list of breakpoints which is more accurate than other methods such as DNAcopy (Table S6), and this degree of accuracy can be critical for several applications.

For example, to construct a genome alteration print, Popova et al. (2009) require a good segmentation algorithm: “the absence of true breakpoints could significantly alter the GAP pattern.” Annotated regions can be used on SegAnnDB to ensure that the segmentation contains all visible breakpoints.

As another example, the locations of detected alterations can be used to construct predictors when modeling survival outcome. SegAnnDB can be used to construct predictors that agree with an expert’s visual interpretation of the data.

An important final example is detecting recurrent alterations in related samples, or excluding germline alterations that also appear in normal samples. There are two approaches: either the model shown on SegAnnDB can be post-processed, or the annotated regions created using SegAnnDB can be used to train and validate another algorithm.

6 CONCLUSIONS

We described the usage and implementation of SegAnnDB, a web-based computer vision system for DNA copy number profile analysis. SegAnnDB improves previous visualization approaches by allowing the model to be interactively updated using annotated regions. In contrast with other mathematical models in the literature, SegAnnDB has no tuning parameters since they are selected automatically using the provided annotations. Overall, SegAnnDB is a useful tool in cancer research that provides accurate annotation of copy number profiles of cancer samples, facilitates interaction between different biologists and bioinformaticians, and allows visualization of several copy number profiles simultaneously.

Annotated regions can be added on SegAnnDB until the displayed model is consistent with an expert’s visual interpretation of a copy number profile. In other words, an expert with enough time can always find a consistent model by adding annotations, since our model always has zero training error with respect to the annotated regions.

However, predicting accurate breakpoints for un-annotated test data was difficult for the models we considered (Figure S5). Developing models that can more quickly achieve better breakpoint detection and copy number calling in un-annotated test data remains an interesting research direction. Although different experts do not always provide consistent annotations (Figure S7), we are also investigating a multi-task learning model that could potentially have better prediction accuracy for each of those experts.

One current feature of SegAnnDB is the ability to plot a random un-annotated chromosome. Instead of a random profile, a profile that is likely to improve model predictions could be shown. We are interested in future research into sampling methods that improve the model faster than random sampling in DNA copy number profile annotation databases.

Finally, we are interested in using visual annotations for modeling copy number changes in next-generation sequencing data. In particular, Teo et al. (2012) show figures with clear breakpoints in real depth and fragment count. Boeva et al. (2012) show plots with clear breakpoints in copy number and B allele frequency. These breakpoints can be visually annotated and saved to a database to ensure that they are respected by any trained models.

Funding/acknowledgements: VB is member of the team “Computational Systems Biology of Cancer,” équipe labellisée par la Ligue Nationale Contre le Cancer. GS is supported by the Annenberg Foundation. We thank Eve Lapouble for help in collecting data. This work was supported by the European Research Council [SIERRA-ERC-239993 to F.B; SMAC-ERC-280032 to J-P.V.]

Conflict of interest: none declared.

REFERENCES


