

Characterization of spinal cord damage based on automatic video analysis of froglet swimming

Sebastián De Vidts*, Emilio Méndez-Olivos**, Miriam Palacios**, Juan Larraín**,
Domingo Mery*+

* Department of Computer Science, School of Engineering, P. Universidad Católica de Chile

** Center for Aging and Regeneration, Facultad de Ciencias Biológicas, P. Universidad
Católica de Chile

+ Corresponding author

Summary

Automatic video tracking system to measure froglet swimming and characterize the degree of spinal cord damage after spinal cord injury.

Abstract

Xenopus laevis frogs are a widely used organism to study aspects of modern biology (Harland and Grainger, 2011). Its central nervous system is particularly interesting, because in certain stages of metamorphosis the spinal cord can regenerate after injury and recover swimming. With this in mind, automatic gait analysis could help evaluate the regenerative performance by means of a method that automatically and quantitatively establishes the degree in froglets' limb movement. Here, we present an algorithm that characterizes spinal cord damage in froglets. The proposed method tracks the position of the limbs throughout videos and extracts kinematic features, which posteriorly serve to differentiate froglets with different levels of damage to the spinal cord. The detection algorithm and kinematic features chosen were validated in a pattern recognition experiment in which 90 videos (divided equally in three classes: uninjured, hemisected and transected) were classified. We conclude that our system is effective in the characterization of damage to the spinal cord through video analysis of a swimming froglet with a 97% accuracy. These results potentially validate this methodology to automatically compare the recovery of spinal cord function after different treatments without the need to manually process videos. In addition, the procedure could be used to measure the kinematics and behavioral response of froglets to different experimental conditions such as nutritional state, stress, genetic background and age.

Keywords: froglets, tracking, swimming, video processing

Introduction

Xenopus offers multiple advantages as a model organism to study development, regeneration, behavior and evolution. Some of the advantages it presents for experimentation are: 1) it is easy to breed and generate hundreds of animals developing synchronously at a low cost

compared to rodents; 2) the availability of genomic resources and methods to study gene function and availability of genomic resources and 3) the possibility to perform tissue transplantation (Harland and Grainger, 2011). *Xenopus* frogs are particularly attractive to study central nervous system (CNS) regeneration because they swim at certain developmental stages when they are able to regenerate the spinal cord after injury and recover the ability to swim. This ability is progressively lost throughout metamorphosis resulting in non-regenerative froglets and adult frogs that are no longer able to recover after spinal cord injury (SCI) (Gaete et al., 2012; Muñoz et al., 2015; Edwards-Faret et al., 2017). The response to SCI in froglets has shown similar molecular, cellular and clinical hallmarks compared to other animals such as rodents and humans (Lee-Liu et al., 2014; Lee-Liu et al; 2017) making this a useful model organism to make studies that then can be informative to understand SCI in humans. In addition, it is quite simple to achieve complex locomotor conditions like paraplegia (with both posterior limbs paralyzed), or hemiplegia (just one posterior limb is paralyzed) two types of lesions that are also observed in humans.

SCI in mammals, including humans, leads to permanent paralysis (paraplegia and quadriplegia) including loss of bowel, bladder and sexual functions, chronic pain and autonomic dysreflexia, among other symptoms. Currently, no efficient treatment for spinal cord regeneration has been established, however, using *Xenopus* as a model organism could be of great help for this. The development of a method that can measure froglets' movement could quantify the effects of new treatments and the recovery of injured animals. In addition, this method could be used to study other behaviors in this classic model organism. For the development of this method, kinematics analysis emerges as an invaluable tool. Gait analysis has been used to study many species, including horses (Barrey 1999), elephants (Hutchinson et al., 2006) and hummingbirds (Warrick et al., 2005). Gait analysis allows the evaluation of pathological conditions in humans (Abbass and Abdulrahman 2014) as well as in other animals (Swanson et al., 1998, Roxana et al., 2007). The swimming of tadpoles and froglets has been analyzed before with kinematics approaches in order to understand the transition from an axial-based swimming to a limbed propulsion (Combes et al., 2004). These approaches have been done for total distance covered or kinematics analysis by drawing the body outline of the animal, making this method very inefficient and laborious and also is not able to measure more complex parameters such as coordination, or compare slight changes in movement in different experimental conditions.

The benefits of using automatic gait analysis for understanding SCR are analysis of a larger number of videos and during longer periods of time, less error than manual tracking and automatic tracking of all limbs simultaneously. In this way, a model-based visual tracking algorithm has been developed for zebrafish (Fontaine et al., 2008) and the fruit fly (Fontaine et al., 2009) both of which achieved a more in-depth understanding of the animal's movement. For example, subtle differences in swimming between normal and mutant zebrafish were measured. An advantage of processing large amounts of videos is the ability to extract efficient features that summarize the information. Features can then be used to compare one or more videos. By using pattern recognition techniques, it is possible to determine which features discriminate best between two or more behaviors.

Here we present an algorithm that analyzes swimming videos and automatically characterizes the level of damage to the spinal cord of a *Xenopus laevis* froglet. From a dorsal view it measures the position of each limb every frame and then analyzes their movement throughout the video. This information is then summarized into four kinematic features that are validated using pattern recognition in a classification experiment. Even though both fore- and hindlimbs

are tracked by the algorithm, the analysis was centered on the latter (Combes et al, 2004). In the first section we describe the detection algorithm for the froglet and its limbs. Subsequently, we define four kinematic features derived from expected behaviors. And finally, we validate the ability of the program to classify 90 videos into three levels of spinal cord damage: uninjured, hemisected and transected animals. The development of an algorithm as a tool with discriminative or “diagnostic” capacity could be of great aid to screen for drugs that have potential beneficial effects on spinal cord regeneration. Here we trained the algorithm with previously well-known conditions using training data and showed that after learning the algorithm is able to classify blindly between different groups. This algorithm is a great improvement in order to quantitatively assess any improvement in spinal cord regeneration after a given treatment. This algorithm would allow the detection of slight improvements in swimming recovery compared to the current methods developed in our own lab that only allows to measure the distance swim by froglets (Gaete et al., 2012; Muñoz et al., 2015), a parameter that is too gross to allow the detection of small effects when screening for potential new treatments.

Materials and methods

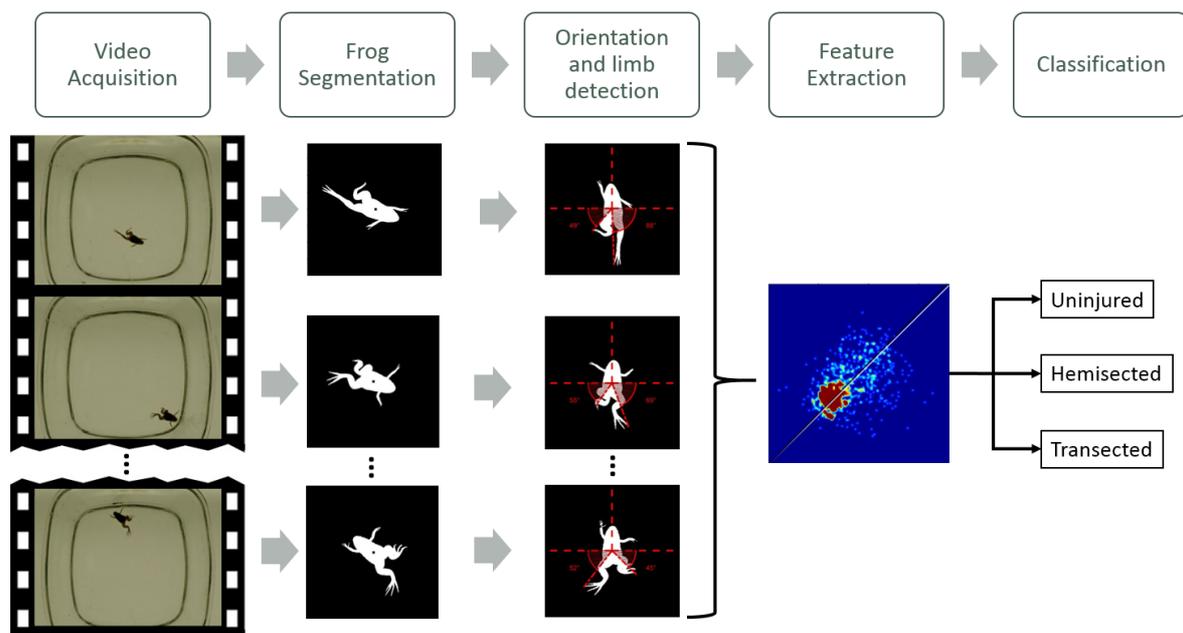


Figure 1: Diagram illustrating the five-step process for each video. The whole video is processed frame by frame and then summarized in the feature extraction phase. Finally, the video is classified into one of the three levels of spinal cord damage.

We used the supervised methodology of pattern recognition (Bishop, 2006). This methodology uses the knowledge of an expert to train a model. In our case, we know how injured are the froglets and the idea is to have an algorithm based on the trained model that can predict the level of damage by analyzing a video without the intervention of the expert. A typical computer vision system based on pattern recognition follows a five-step schema: 1) image acquisition, a digital image or video of the object under test is taken (sometimes resized) and stored in a

computer, 2) preprocessing, the digital image is improved in order to enhance the details, 3) segmentation, the image of the object of interest is identified and isolated from the background of the scene, 4) feature extraction/selection, significant features of the object are quantified and 5) classification done by a classifier (Bishop,2006). The five steps must be designed for a specific application using training data. In our case, the information is taken from videos of swimming froglets that are processed and segmented in order to detect their anatomy. Afterwards, certain features are extracted from the movement of the froglets that are used to determine the spinal cord damage category.

Following and adapting this scheme, we have divided our algorithm into five blocks that are shown in Fig. 1. As we mention in the introduction, the objective of our experiment is to correctly classify all 90 videos into the three spinal cord damage categories (uninjured, hemisected and transected). The algorithm first identifies the froglet in each frame and segments its shadow for the next step. This image is then oriented so as to always be looking upwards while in the dorsal view and detects the position of its limbs with respect to the center of gravity. The whole process outputs four kinematic features (synchronization, symmetry, range of the angles for the right and left foot) and a 2D heatmap which shows the angles of both legs of the froglet moved throughout the video. Then, a classifier is used to predict to which of the three previously defined categories of spinal cord damage the froglet belongs to. The following subsections describe in greater detail each part of the algorithm.

1) Spinal cord injury of *Xenopus laevis* froglets

Xenopus laevis froglets were used at 2 weeks after reaching post-metamorphic stage 66, at which point they measured approximately 1 cm wide and 1.5 cm long (Figure 2). Spinal cord transection was performed as described in Edwards- Faret et al., (2017). Animals were anesthetized by immersing them in 0.02% MS222 0.2 g/L) for 10 minutes and placed on a gauze over an inverted petri dish to avoid any spinal cord damage. Using microdissection scissors an incision was made at the level of the sixth vertebrae and laminectomy was performed followed by full spinal cord transection or only the right half of the spinal cord was transected for the hemi-section. After injury, muscle and skin were put together using forceps to allow closure and healing of the incision. After surgery, both hindlimbs were paralyzed in transected froglets, whereas only the right hindlimb was, paralyzed in hemisected froglets.

Careful postoperative manipulations were carried out as described (Edwards-Faret et al., 2017). After surgery, the froglet was maintained in the gauze of operation and transferred to plastic tanks containing 200 ml of 0.1X Barth¹ solution with 1X antibiotics, with no more than six froglets per tank. To allow breathing, the solution must not cover the froglets' heads. The recovery of anaesthesia occurred 5-10 min after the surgery, when animals begin to move their upper extremities and then are transferred to the larvae/froglet room at 20-21°C. We started feeding the animals one day after surgery with pellets Nasco Frog Brittle and the pellets were brought to the forelimbs with a Pasteur pipet to help them eat. The peeling skin is removed with forceps under a dissecting microscope, once a week. It is necessary to change the Barth x1 solution with antibiotics every other day and clean the tanks

¹ It is not possible to specify a unit of measure; BARTH is the standard used with frogs.

A total of 90 froglets were recorded, 30 froglets were used for each condition. The swimming test was performed three hours after surgery once the animals had completely recovered from anesthesia. All experiments with froglets and the procedures used in this work have been approved by the “Comité Ético Científico para el Cuidado de Animales y Ambiente” at the Pontificia Universidad Católica de Chile. Number of identification: 150408002.

2) Video Acquisition

Uninjured, hemisected and transected froglets were recorded in the dorsal view using a GoPro Hero5 Black camera that was mounted on a tripod as illustrated in Fig. 2. Animals were maintained in a glass container with 500 mL of 0.1x BARTH solution (pH 7.6) supplemented with penicillin-streptomycin at room temperature (25°C). The froglets were placed on the glass container using a spoon and filmed one-by-one for one minute after carefully touching the forelimbs with a Pasteur pipette (swimming stimulation). The data was acquired at 120 color frames per second with a resolution of 1,920×1,080 pixels for each color channel (red, green and blue). The length of the videos is approx. 1 minute.

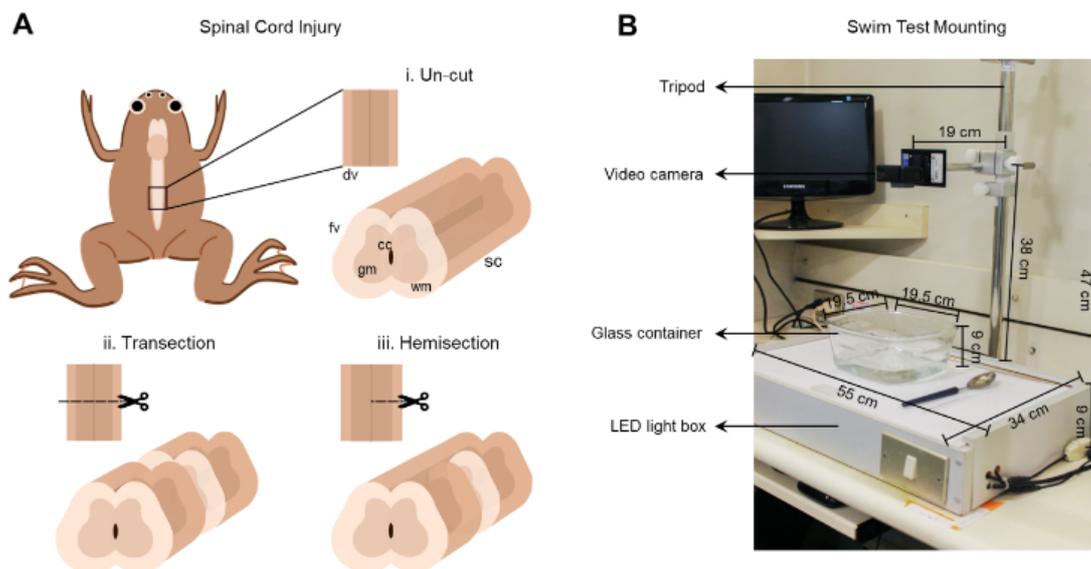


Figure 2. (A) Illustration of spinal cord injuries in the 6th vertebrae of *Xenopus laevis* froglets in the dorsal and frontal views. (i) uninjured, (ii) transected and (iii) hemisected animals. (B) Picture of the setup used to perform movies of animals. The tripod where the video camera is placed, a box internally illuminated with LED lights, a glass container with Barth solution and a spoon and a Pasteur pipette for swimming stimulation of froglets. fv: frontal view and dv: dorsal view of the spinal cord. sc: spinal cord, gm: grey matter, wm: white matter and cc: central canal.

3) Froglet Segmentation

• Yolo Deep Learning Network

The first step of the segmentation is based on a deep learning algorithm called Yolo-v3 (You Only Look Once) version 3 (Redmon and Farhadi, 2018). A subset of 1800 frames were selected randomly from 30 videos from our dataset (10 videos of each spinal cord damage category). In addition, we manually detected the froglets in each video in order to generate

the training dataset. The model was trained for 2000 iterations and validated visually with images from the other 60 videos.

- **Area range calibration**

During the identification and tracking, the algorithm filters out objects that have an area too large or too small to be the froglet (to prevent the detection of other objects located in the video). In order to do this, the area of the froglet is measured. For each video six random frames are extracted, the froglet is then detected in each image using the Yolo Network previously described. By using a color threshold and the (x,y) coordinates of the detection in the frame, the area of the froglet is measured in each of the frames of the video. The calibration area is defined as the median of the measured areas of the images. Then the froglet's area range is defined as 10% larger and smaller than the calibration area. Multiple thresholds were tested using our video database and 10% was observed as difference of area of the froglet throughout a video.

- **Froglet detection**

The algorithm processes one every two frames to reduce processing time (in our experiments, this rate still maintained accurate results), while maintaining accurate results. The first step was a threshold in the blue channel, which of the three color channels has a larger difference between the froglet and the background creating a better segmentation. This outputs a binary image where the froglet was seen as white and the background as black. The area of each object in the image were measured, filtering those whose area fall outside the range defined in the calibration section. Of the remaining objects, the one closest to the previous frame detection is chosen as the froglet. A fixed size window of the frame is extracted, the coordinates of the center of the frame are used for the detection of the next analyzed frame.

4) Orientation and limb detection

- **Orientation**

The binary segmented image is virtually eroded in order to eliminate the limbs, leaving only the body. Virtual erosion is accomplished by thinning the outer pixels of the object in the binary image. The orientation is measured using the Matlab function 'Regionprops: Orientation' (MathWorks, 2017). The image is then oriented so as to always be looking upwards while in the dorsal view and detects the position of its limbs with respect to the center of gravity. By checking the number of pixels in the bottom half versus the top half of the image (because of a difference of pixels between both halves) one can correct that the froglet is "looking up" and not "looking down".

- **Limb detection**

The limb detection algorithm (Fig. 3) starts with a binary image of the previous step. By subtracting an eroded image from the original one, the limbs are extracted. Both limbs and body of the froglet are then dilated to find the area where they intersect. The dilation adds pixels to the boundaries. This area is defined as the "joint" of limbs and body. Using Matlab function `bwmorph: skel` the "morphological skeleton" is extracted from the segmented image. Using `bwmorph: endpoints`, the endpoints of the skeleton are extracted. These "points of interest" are considered as the hands or feet of the froglet. The information of both

branches is combined and divided into four quadrants, one for each limb. For each quadrant the furthest point of interest from its joint is selected as the foot or hand of the froglet

After the limbs are detected in a frame, the angle-pair of the limbs (α_R, α_L) is computed (Fig. 4). The angle-pair is defined as the two angles between the right and left foot and the horizontal line. This makes it easier to compare the movement of both legs. We define a 2D histogram as a matrix of 180×180 bins, where each axis represents the angles α_R and α_L , and each bin (with a resolution $1^\circ \times 1^\circ$) counts how many frames of the video have the legs in the angle-pair defined by the bin. To show which angle-pairs were more common to the froglet during the video, the histogram is normalized between the maximum and minimum bin value and visualized as a heatmap. The most common angle-pairs are shown with a red color, while a non-detected angle-pair throughout the video is colored blue.

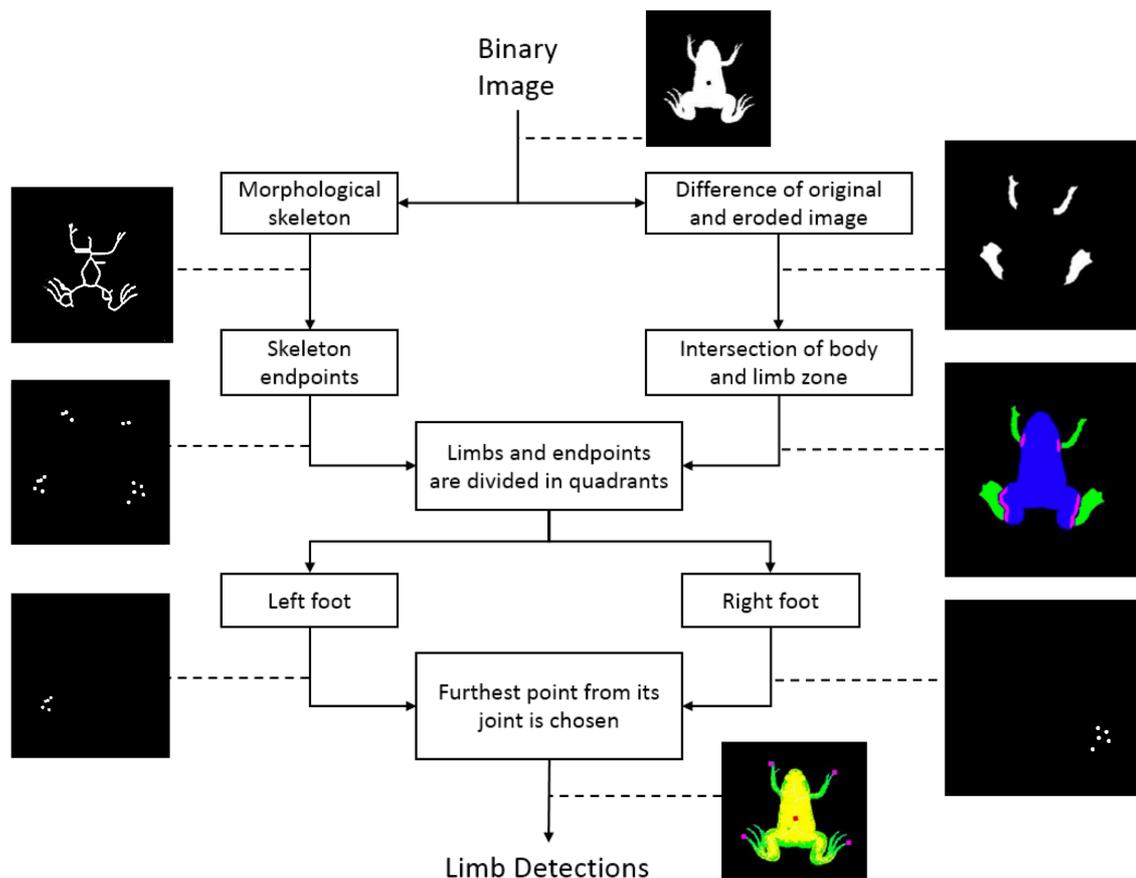


Figure 3: Diagram showing the steps to detect the froglet's limbs. The algorithm detects limb zones, joint zones and endpoints for each of the four quadrants. Each limb will be defined by the endpoint furthest from its joint.

5) Feature Extraction

Information from the videos are summarized in four kinematic features. The contributions of these features are three-fold: 1) They are highly discriminative, that means in this classification problem, they achieve a high level of variance between classes and maintaining a low level of variance within a class. 2) They are derived from expected visual behavior with an easy interpretation. 3) They can be easily computed using a mathematical formula. In addition,

these features allow a rapid understanding of which parameters of froglet swimming are altered and which can be improved with different treatments.

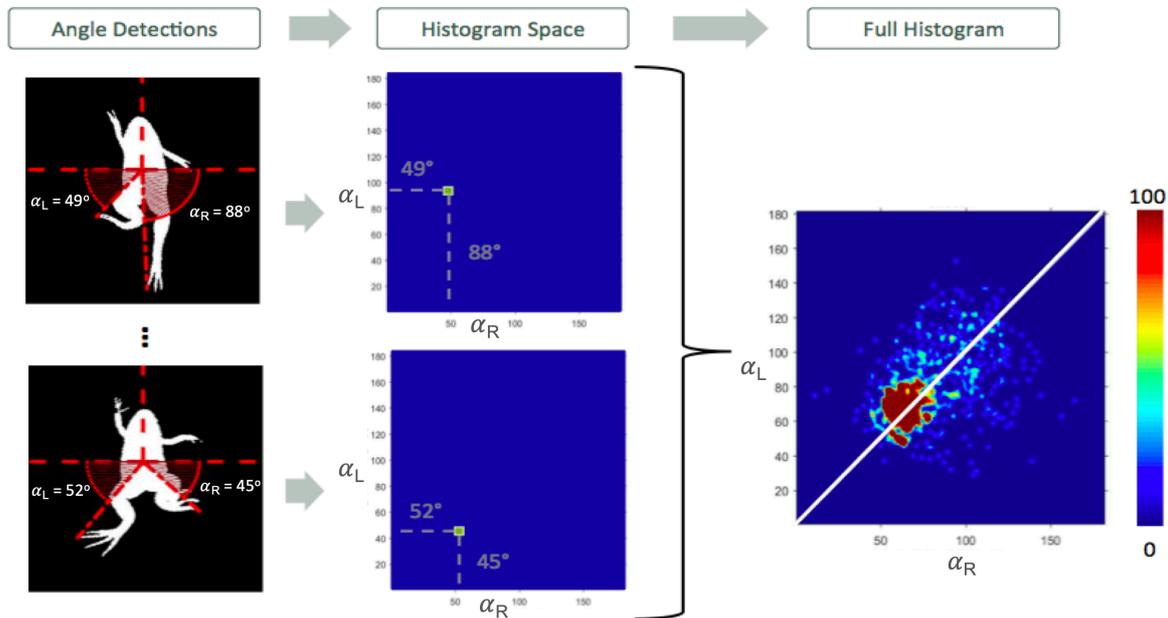


Figure 4: Example of a 2D histogram is constructed and outputted by the algorithm. In the examples, the angle-pairs (α_R, α_L) are $(49^\circ, 88^\circ)$ and $(52^\circ, 45^\circ)$. For these frames, the bins $(49, 88)$ and $(52, 45)$ of the 2D histogram are added by one. The histogram is represented at the right as a heatmap with a color scale: the reddest value is associated to a most common angle-pair throughout the video, while the blue is associated to an angle-pair not seen throughout the video. The white line represents perfect symmetry, meaning that both feet are horizontally mirroring.

• Synchronization

The first feature referred to the synchronization of hind limb movement. It captures the synchronization of the feet while swimming. Three types of synchronization are presented to illustrate the observable behaviors: (i) *Synchronized*, fore and hindlimbs from the right and left move forward or backwards at the same time. (ii) *Independent*, only one of the hindlimbs moves while the other stays still. (iii) *Inverted*, one hindlimb moves forward while the other moves backwards and vice versa. By observation, the healthy froglet should swim synchronized, while a hemisected froglet should move only one hindlimb and a transected froglet cannot move both hindlimbs, but the movement of the pelvis is identified as an inverted movement of the feet by the algorithm.

Synchronization was measured as the correlation between the angles α_R and α_L (Fig. 4). If the correlation between the feet was close to 1 then we determined that both feet were synchronized in their swimming. On the other hand, if the correlation was close to -1, the movement of the feet was inverted. And, if the correlation was close to 0 then the feet were moving independently of each other, which potentially means the froglet is moving one foot while leaving the other still. As the froglet will have slight unsynchronized movements, the expected Synchronization coefficient for synchronized swimming will be closer to 0.5, while an inverted movement will be close to -0.5.

- **Symmetry**

The second feature is symmetry in the feet's strokes. In the average, stroke of a healthy froglet, we observe that both feet should move in a similar arc. On the other hand, a hemisected froglet will move the healthy foot in a larger arc than the damaged side. In our observations, the transected froglet has little movement of its legs, but said movement will tend to be symmetrical.

Stroke symmetry can be measured with a first order linear regression. The linear regression finds out the relationship between the angles α_R and α_L by finding the slope β_0 which best solves the equation:

$$\alpha_R = \beta_0 \alpha_L \quad (1)$$

where β_0 is the symmetry coefficient. If both feet move symmetrical to each other, β_0 will be close to 1, but if the right foot moves half the arc than the left foot, then β_0 will be close to 0.5.

- **Range**

The last two features are the range of angles α_R and α_L . A healthy froglet has a relative larger range of angles of both of its legs, while a transected one will have a relatively smaller range. The non-injured leg of a hemisected froglet will fall somewhere in-between, having more movement in its uninjured side than the damaged side. To measure this, the detections of each foot are smoothed using a bivariate kernel density estimator (KDE) and a gaussian convolution (Botev, 2010) which is a good solution for noise reduction. A threshold is applied to the result and an area of movement of the foot is obtained. The highest and lowest point is calculated from that area and the range of movement is the arc made by both points and the center of gravity of the froglet.

6) Classification

The final step of the algorithm is a classifier. In our case, the idea of the classifier is to assign the froglets to one of the three classes: uninjured, hemisected and transected. The classifier takes the extracted features of a froglet and assigns the froglet to one of these classes. The extracted features are arranged in a vector of n elements that are represented in a feature space of dimension n . For example, if we extract the features $F1$ (synchronization) and $F2$ (symmetry), the feature space is a 2D space, where (x,y) axes corresponds to the variables $(F1,F2)$, respectively. Thus, a froglet was represented as a point in this 2D space as illustrated in Fig. 9. In order to design a classifier, the feature space is divided into three regions, that we called 'decision regions', one for each class. This step is called training, in which a model that finds the decision regions is estimated. After training, if we want to classify a new froglet, we extract the proper feature vector, a point in feature space and classify it according to the decision region to which the point belongs.

It is worth mentioning that when designing a classifier, two stages are taken into account: training and testing. On the one hand, in training stage, training data (froglets for training purposes) are used to define the 'decision regions'. Typically, a classifier defines the decision regions according to the concept of similarity: features that are similar are assigned to the same class. On the other hand, in the testing stage, testing data (other froglets that were not used in the training stage) are used to evaluate the performance of the classifier. That means,

features are extracted from the testing froglets and classified using the trained model. In the evaluation, the accuracy defined as the ratio of test froglets correctly classified is computed. In our experiments, we tested simple classifiers because of the small number of samples that was available. The classifiers that we used are: minimum distance, linear discriminant analysis (LDA), k-nearest neighbors (KNN) with 1, 3, 5 neighbors; and more complex ones such as: quadratic discriminant analysis (QDA), a shallow neural network, Mahalanobis distance and support vector machine (SVM) in both its linear and radial basis function kernel (RBF) (Bishop, 2006). For this end, we use the implementation of Balu Matlab Toolbox (Mery, 2011).

The performance of the classification is evaluated using cross-validation (Mitchell 1997). In our experiments, the data are divided into 10 folds (independent groups), because it has become the standard method in practical terms (Witten and Frank, 2005). That means, 90% are used for training (81 frogs) and 10% for testing (9 frogs). This experiment was repeated 10 times interchanging training and testing data to evaluate the stability of the classifier. Then, when training was performed, the samples that were initially removed could be used to test the performance of the classifier on these test data. For each time, the performance defined as the rate of samples correctly classified is computed as η_i , for $i=1\dots 10$. Thus, we evaluated the generalization capabilities of the classifier by testing how well the method classified samples that had not been already examined. The estimated accuracy, η , is calculated as the mean of the 10 percentages of the true classifications that are tabulated in each case: $\eta = (\eta_1 + \dots + \eta_{10})/10$.

To validate both the algorithm and kinematic features, the performance of different pattern recognition classifiers was measured to see how much variance exists between the three categories of spinal cord damage. A high accuracy of classification shows that the proposed system of algorithm and features can be used to differentiate levels of spinal cord damage, being useful to compare damage level from different froglets.

Implementation

The algorithm was implemented in Matlab except for Yolo approach, which was done using the original implementation given by the author (Redmon and Farhadi, 2018). The average processing time is 1.12s/frame in a 3.4GHz Intel i7 processor, which takes about 1 hour of processing for each 1-minute video. The output of our code is a list of the detection information for each frame, the classification of the froglet (one of the three spinal cord damage categories) and the four kinematic features which can be used to further study the swimming performance. Code and videos are available².

Results

Kinematic features

For each video processed, a heatmap of the angles for the right and left hind limb were obtained and plotted in a 2D histogram (Fig. 5 and 6). In the example of Fig. 5 an uninjured froglet shows a positive correlation with the algorithm (0.636), instead of a negative one for transected swimming (-0.768). So uninjured froglet are detected with a synchronized

² See <https://domingomery.ing.puc.cl/material/>

swimming, according to the swimming description made by Combes et al., 2004. Meanwhile in same figure, the example of a hemisected froglet the correlation is near to zero (0.082), indicating an independent or non-synchronized movement. On the other hand, examples of symmetry of swimming are shown in Fig. 6. There for an uninjured froglet as the coefficient is equal to 0.98, close to 1. But for a hemisected froglet the coefficient is equal to 0.55; indicating a lack of symmetry because of the less stroke movement of the right hind limb which was damaged, compared to the left hind limb.

All results of the videos were plotted on Fig. 7 (A) for the two kinematic features of symmetry and synchronization. As is expected, healthier froglets were identified by the algorithm by a more synchronized swimming and resulted with a positive symmetry coefficient (F1). While most transected froglets resulted with a negative correlation of synchronization, indicating inverted movement, hemisected froglets differ on their synchronization results with a range between -0.2 and 0.5, so not all of them identify with an independent movement of their hind limbs for the algorithm. However, the swimming of hemisected froglets differ from uninjured and transected since most of them obtained a coefficient of symmetry (F2) less than 0.9. Furthermore, both transected and healthy froglets have a similar level of symmetry in their swim, near to +1: in the case of the uninjured froglets this is explained because it makes similar strokes with both left and right hind limbs and in the case of the transected animals because it has little movement in both of its hind limbs. The hemisected froglet clearly shows lower movement of its right hind limb in comparison to its left one. This is expected, as all hemisected frogs were damaged in the spinal cord's right side.

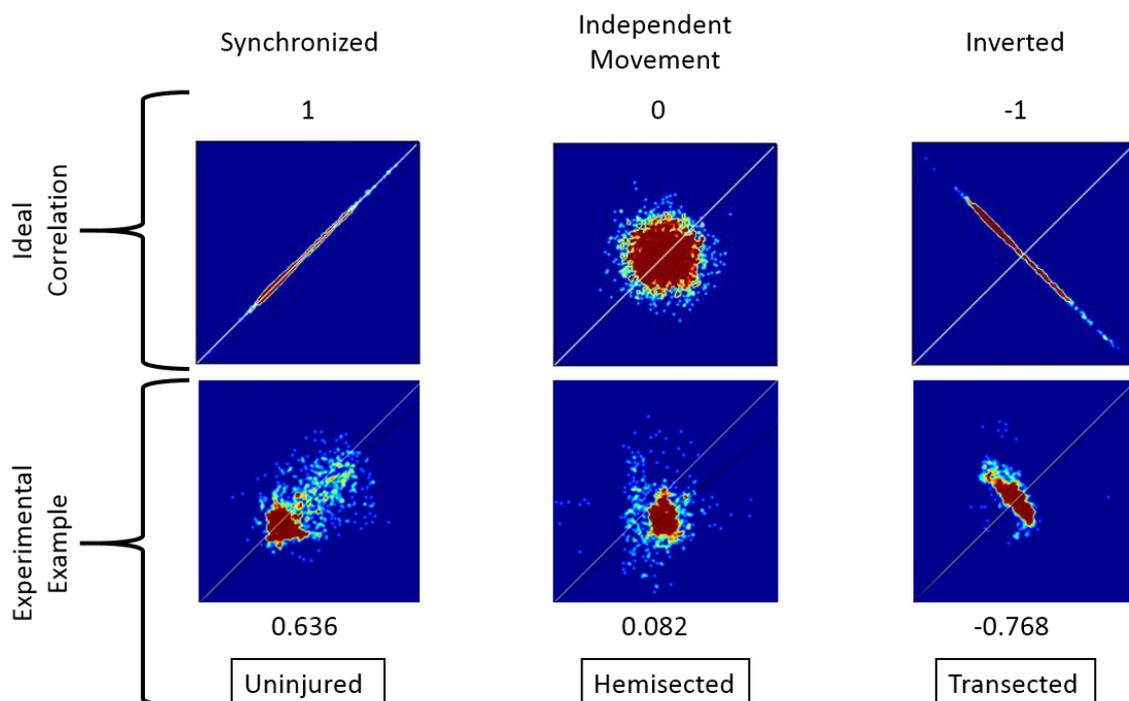


Figure 5: Synchronization variance in 2D heatmaps and the expected output of each behavior. One example is given of each type of froglet compared to the ideal correlation of each of the three types of synchronization

previously defined: Synchronized, independent and inverted. Rows show respectively: type of movement, ideal correlation, ideal distribution, one experimental example, measured correlation and original class.

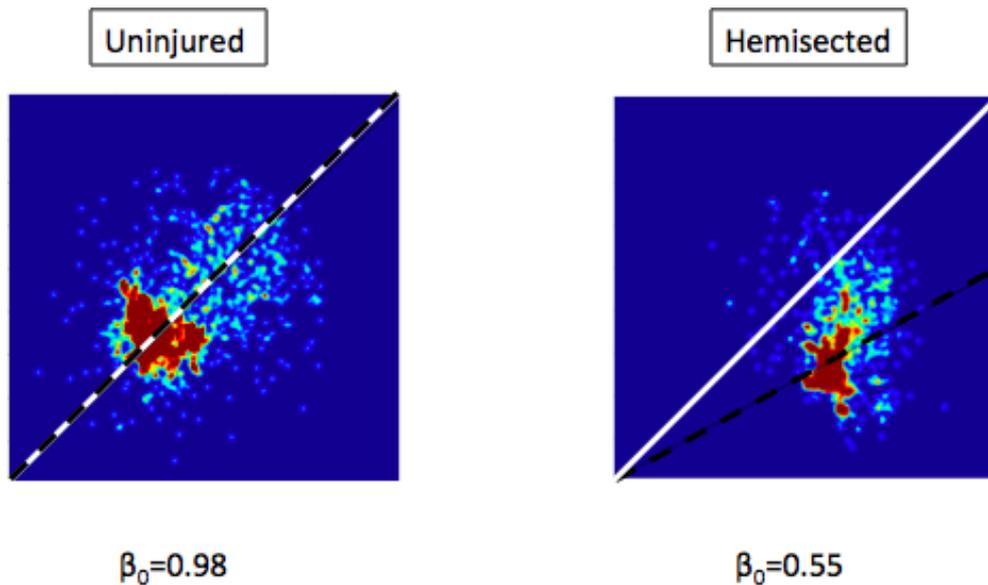


Figure 6: Symmetry variance in 2D heatmaps. Two experimental examples to illustrate how a difference in symmetry is reflected the heatmaps. The white line shows perfect symmetry, while the black one shows the corresponding line of the estimated slope β_0 . We can see that in the healthy froglet both black and white lines are almost the same, giving a β_0 almost equal to 1. On the other hand, in the hemisected froglet the right foot tends to move almost half of the left foot's stroke, making white and black lines differ significantly, thus showing a lack of symmetry.

As expected, uninjured froglets have a relatively high range of movement for both feet, while transected have a small range (Fig. 7). Hemisected fall in between, having a similar range to a healthy froglet in its left hind limb, while the right is similar to a transected froglet. In Fig. 8 two examples are shown, where a healthy froglet reaches a 97° of movement meanwhile a transected froglet reaches a 38° of movement. There is more confusion in the classification using range features than using synchronization and symmetry features.

Classification results

In our experiments, we try using both simple classifiers (e.g. minimum distance and linear discriminant analysis) and more complex ones (e.g. SVM and neural networks).

Each classification experiment was performed using cross validation with 10 folds as described in Materials and Methods and repeated 10 times for each classifier used (Table 1). Using only two features, synchronization and symmetry, common classifiers as KNN with 3 and 5 neighbors get over 90% accuracy, not considerably lower than the 93.9% accuracy with a neural network. Using only the right and left foot range almost all classifiers get close to an 80% accuracy. Finally, the best result is obtained combining all 4 kinematic features in a LDA getting a 96.6% accuracy.

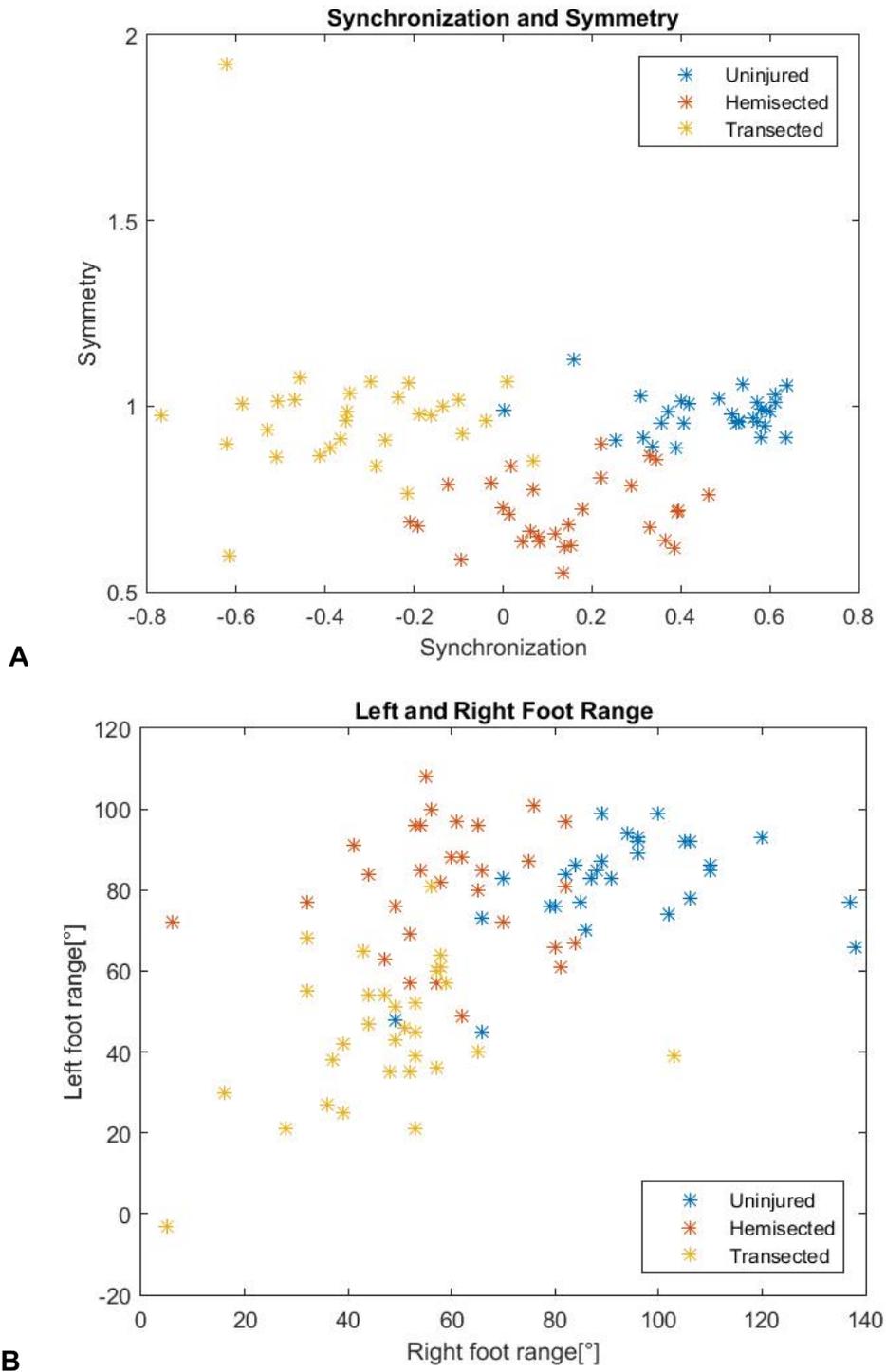


Figure 7: Examples of distribution in the feature space of the 90 videos using two kinematic features. They are colored with their original class (uninjured, hemisected and transected). In panel (A) the features synchronization (F1) and symmetry (F2) are shown, whereas in panel (B) features of right and left foot angles range (F3 and F4) are shown.

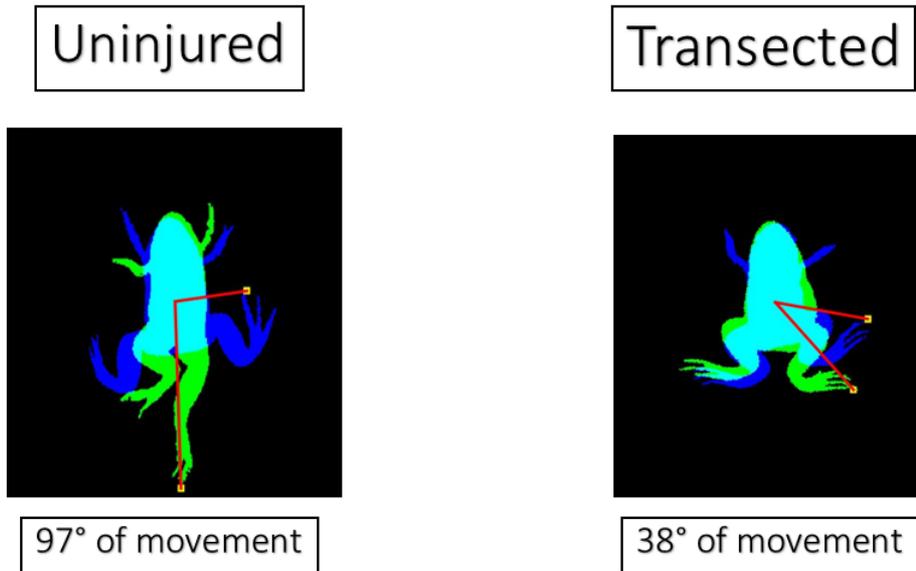


Figure 8: Two examples of the Right Foot Range (F3) measured. The blue shadow shows the moment where the froglet has its foot the highest, while the green is the lowest. The range is defined as the arc between the highest and lowest position. The healthy froglet has a relatively large range of movement while the transected one only moves its pelvis, having a relatively small range of movement.

Table 1: Average accuracy (mean) and standard deviation (Std) of different combinations of classifiers and features. Each experiment was done using 10 group cross validation and repeated 10 times.

Features →	Accuracy [%]					
	All four features		Symmetry and Synchronization		Right and left foot range	
Classifier ↓	Mean	Std	Mean	Std	Mean	Std
KNN-1	78.70	2.47	87.00	1.66	72.00	2.03
KNN-3	76.10	1.58	91.30	0.82	77.30	1.18
KNN-5	77.20	1.43	92.60	0.35	78.60	1.41
LDA	96.60	0.35	93.60	0.59	80.90	0.70
QDA	93.30	1.72	92.70	1.10	79.10	1.50
Neural Network	95.10	1.55	93.90	1.31	79.90	0.47
SVM Linear	94.70	1.05	90.70	1.46	81.80	0.70
Minimum Distance	78.20	1.05	89.90	1.02	79.00	0.88
SVM RBF	91.20	1.83	90.40	1.66	78.00	2.17
Mahalanobis	91.90	1.41	91.20	0.97	79.10	0.94

The designed classifiers divided the space of features in decision boundaries (Fig. 9). A classifier like the KNN-1 will have a more complex division as it bends to capture all the data, but the result won't generalize as well to new data. The neural network on the other hand, has

straight lines and clearly defined regions, leaving out some videos as exceptions in order to have a more generalized division.

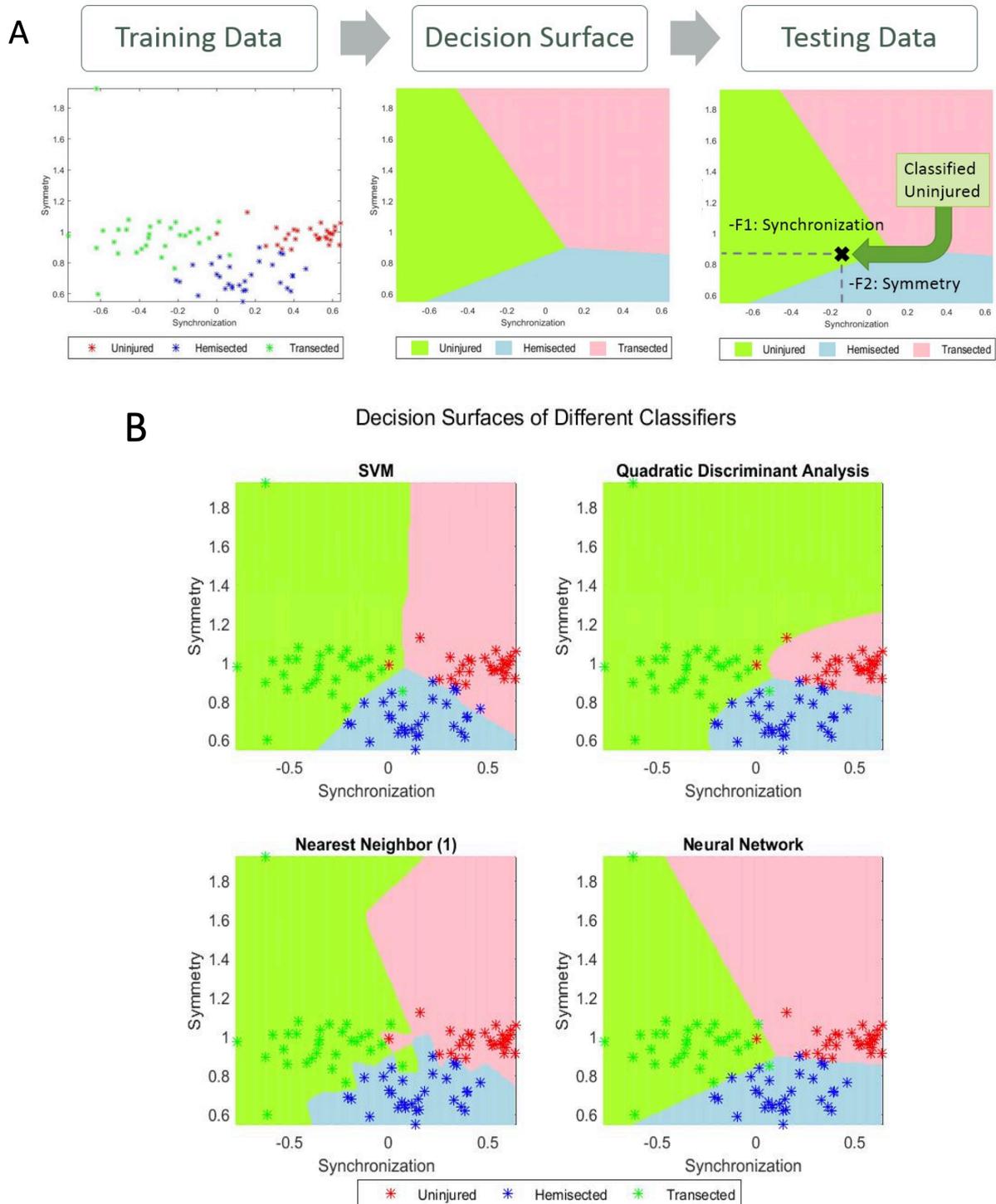


Figure 9: Examples of the output of a classifier plotting all 90 videos using two features: Synchronization (F1) and Symmetry (F2). (A) Diagram of a classification process. Labeled training data is computed to a decision surface of the feature space. This is a mathematical formula that depends on the type of classifier. New data can be tested and classified depending where it falls in the decision surface (B) Different classifiers and how they divide the feature space in decision boundaries.

Discussion

The main contributions of the paper are twofold. On one hand, we define four kinematic features (synchronization, symmetry, range of right foot and range of left foot) that can be used to characterize the swimming behavior of the froglets. On the other hand, we validate how relevant are these features in a classification problem, where the proposed features can be used to determine the level of the spinal cord damage of the froglets and the identification of new treatments for improving spinal cord regeneration.

We present a fully automatic detection algorithm that can characterize the level of spinal cord damage of a froglet's spinal cord by tracking its hind limbs movements while it swims in laboratory conditions. A one-hour processing time for every video may seem lengthy, this process is completely unsupervised, not using the researcher's time. On the other hand, the alternative of having a human segmenting and tagging each limb will be considerable slower than 1.12s per frame. This will let researchers process larger amounts of videos without the need to manually label each limb in each frame. In addition to the detections in less time, we have also provided 4 kinematic features which can be used to summarize the results and variance between healthy and injured froglets. An example would be a froglet showing signs of recovery after receiving treatment. The simple setup makes it easy to reproduce the conditions of the experiment with a common equipment.

Our approach has a high accuracy in classifying *Xenopus laevis* froglets in three categories of spinal cord damage using four kinematic features. It does not lose much accuracy by only using two of them (synchronization and symmetry, by being able to show the results in a 2D graph. As the proposed features create a high difference between the classes, more complex classifiers get similar results to simpler ones.

Also, this system could be used to compare swimming of froglets where the independent variable is other than spinal cord damage, like thermal stress (Padilla et al., 2019), genetics, etc. Our algorithm does not evaluate the Z axis, making it unusable for a question about tridimensional behavior, which is considered as a complex behavior in term of mobility and locomotion (Macri et al., 2017). We do not evaluate this behavior because induce unnecessary stress for breathing in the lesioned froglets, that are not able to reach the surface when they need to.

It is worth mention that some videos fall outside of their class distribution, making them prone to be erroneously classified. These errors come from two different sources: the first one being minor errors of tracking and measurement of said kinematic features (this error is produced by our algorithm); the second source are uninjured froglets which are turning while swimming. When a froglet tries to turn to avoid the borders of the container, it uses only one of its legs to swim forward. Thus, the algorithm interprets this as independent movement. If the uninjured froglet turns the whole video, then the algorithm will wrongfully interpret said movement. This could be corrected by analyzing only frames where the froglet is swimming far from the boundaries.

The algorithm depicted here could have potential application in order to screen libraries of compounds to identify new drugs that can be used to improved functional recovery after spinal cord injury. Having this algorithm should allow the measure of small improvements on froglet

swimming capacity when comparing froglets treated with those compounds with controls froglets. Identifying compounds that improve spinal cord recovery in froglets could be a first and easy step to have candidate drugs that afterwards can be tested in more expensive model organism such as mouse and/or monkeys as another step before performing clinical trials in humans.

Acknowledgements

We thank members of the Larraín Lab. for their support, especially our dedicated former animal caretaker E.E.M.-O. who was a CONICYT PhD fellow.

Competing interests

No competing interests declared.

Funding

This work was supported in part by Fondo de Fomento al Desarrollo Científico y Tecnológico (Fondef) Grant No. ID15I10349, CARE Chile UC-Centro de Envejecimiento y Regeneración (PFB 12/2007) and in part by Fondo Nacional de Desarrollo Científico y Tecnológico (Fondecyt) Grant No. 1161314 from Comisión Nacional de Investigación Científica y Tecnológica (CONICYT), Chile.

References

- Abbass, S. J. and Abdulrahman, G.** (2014). Kinematic analysis of human gait cycle. *College of Engineering Journal (NUCEJ)*. 1616(22), 208–222.
- Barrey, E.** (1999). Methods, applications and limitations of Gait analysis in horses. *Veterinary Journal*. 157(1), 7–22.
- Bishop, C. M.** (2006). *Pattern Recognition and Machine Learning*. New York, USA: Springer.
- Botev, Z. I., Grotowski, J. F. and Kroese, D. P.** (2010): Kernel density estimation via diffusion. *Annals of Statistics*, 38 (5), 2916-2957.
- Combes, D., Merrywest, S. D., Simmers, J. and Sillar, K. T.** (2004). Developmental segregation of spinal networks driving axial- and hind limb-based locomotion in metamorphosing *Xenopus laevis*. *Journal of Physiology*. 559, 17–24.
- Currie, S.P. and Sillar, K.T.** (2018). Developmental changes in spinal neuronal properties, motor network configuration and neuromodulation at free-swimming stages of *Xenopus* tadpoles. *Journal of Neurophysiology*. 119 (3) 786-795.
- Edwards-Faret, G., Muñoz, R., Méndez-Olivos, E., Lee-Liu, D., Tapia, V.S. and Larraín, J.** (2017) Spinal cord regeneration in *Xenopus laevis*. *Nature Protocols*. 12, 372-389.
- Fontaine, E.I., Lentink, D., Kranenborg, S., Müller, U. K., van Leeuwen J. L., Barr, A. H. and Burdick, J. W.** (2008). Automated visual tracking for studying the ontogeny of zebrafish swimming. *Journal of Experimental Biology*. 211, 1305-1316.
- Fontaine E. I., Zabala, F., Dickinson M. H. and Burdick, J. W.** (2009). Wing and body motion during flight initiation in *Drosophila* revealed by automated visual tracking. *Journal of Experimental Biology*. 212, 1307-1323.

- Gaete, M., Munoz, R., Sanchez, N., Tampe, R., Moreno, M., Contreras, E. and Larrain, J.** (2012). Spinal cord regeneration in *Xenopus laevis* proceeds through activation of Sox2+cells. *Developmental Biology*. 356(1), 381.
- Harland, R. M. and Grainger, R. M.** (2011). *Xenopus* research: Metamorphosed by genetics and genomics. *Trends in Genetics*. 27(12), 507–515.
- Hutchinson, J. R.** (2006). The locomotor kinematics of Asian and African elephants: changes with speed and size. *Journal of Experimental Biology*. 209, 3812–3827.
- Lee-Liu, D., Moreno, M., Almonacid, L.I., Tapia, V.S., Muñoz, R., von Marees, J., Gaete, M., Melo, F., and Larraín, J.** (2014) Genome-wide expression profile of the response to spinal cord injury in *Xenopus laevis* reveals extensive differences between regenerative and non-regenerative stage. *Neural Dev.* 9, 12.
- Lee-Liu, D., Méndez-Olivos, E. E., Muñoz, R. and Larraín, J.** (2017). The African clawed frog *Xenopus laevis*: a model organism to study regeneration of the Central Nervous System. *Neuroscience Letters*, 652, 82–93.
- Macri, S., Neri, D., Ruberto, T., Mwaffo, V., Butail, S., & Porfiri, M.** (2017). Three-dimensional scoring of zebrafish behavior unveils biological phenomena hidden by two-dimensional analyses. *Scientific reports*, 7(1), 1962.
- MathWorks** (2017): Matlab Toolbox - Image Processing Toolbox, Reference Guide. MathWorks, Inc.
- Mery, D.** (2011), BALU: A Matlab toolbox for computer vision, pattern recognition and image processing. Software available at <http://dmery.ing.puc.cl/index.php/balu>.
- Mitchell, T.** (1997). *Machine learning*. Boston, US: McGraw-Hill.
- Muñoz, R., Edwards-Faret, G., Moreno, M., Zuñiga, N., Cline, H. T. and Larraín, J.** (2015). Regeneration of *Xenopus laevis* spinal cord requires Sox2/3 expressing cells. *Developmental Biology*. 408, 229–243.
- Padilla, P., Ducret, V., Bonneaud, C., Courant, J., & Herrel, A.** (2019). Acclimation temperature effects on locomotor traits in adult aquatic anurans (*X. tropicalis* and *X. laevis*) from different latitudes: possible implications for climate change. *Conservation physiology*, 7(1), coz019.
- Picton, L.D., Sillar, K.T. and Zhang, H-Y** (2018): Control of *Xenopus* tadpole locomotion via selective expression of Ih in excitatory interneurons. *Current Biology*, 28 (24), 3911-3923.
- Redmon, J. and Farhadi, A.** (2018). YOLOv3 An incremental improvement. arXiv:1804.02767.
- Roxana, D., Igna, C., Sabau, M. and Menyhardt, K.** (2007). Kinematic Analysis of the Gait in Healthy Common Breed Dogs. *Bulletin UASVM CN*. 64, 409–414.
- Swanson, C., Young, P. and Ceche Jr, J.** (1998). Swimming performance of delta smelt: maximum performance and behavioral and kinematic limitations on swimming at submaximal velocities. *Journal of Experimental Biology*. 201, 333–345.
- Warrick, D. R., Tobalske, B. W. and Powers, D. R.** (2005). Aerodynamics of the hovering hummingbird. *Nature*, 435(7045), 1094–1097.
- Witten, I. and Frank, E.**, (2005). *Data mining: practical machine learning tools and techniques* (2nd. ed.). San Mateo, CA: Morgan Kaufmann.