FishPose: Identifying fish skeletal elements from X-ray images

A computer vision project

TzuChiao Hung
Department of Developmental Biology
Stanford University
tzuchiao@stanford.edu

Motivation

Our lab studies the evolution of skeletal traits. Measuring lengths of various bones is a fundamental routine. By doing so we have discovered genetic elements that, through regulating relative lengths of different bones, gave rise to diverse forms of organisms[1–3]. Nevertheless, the process itself was repetitive and tedious, which motivated me to automize the task: *given an X-ray image, the algorithm should return the lengths of all skeletal elements of interest*. This requires the algorithm to identify the coordinates of the two ends of interested bones. As a pilot model, FishPose deals with X-ray images of threespine sticklebacks (*Gasterosteus aculeatus*), for our lab has a database of thousands of such images, and the planar nature of the fish body plan simplifies this incipient project.

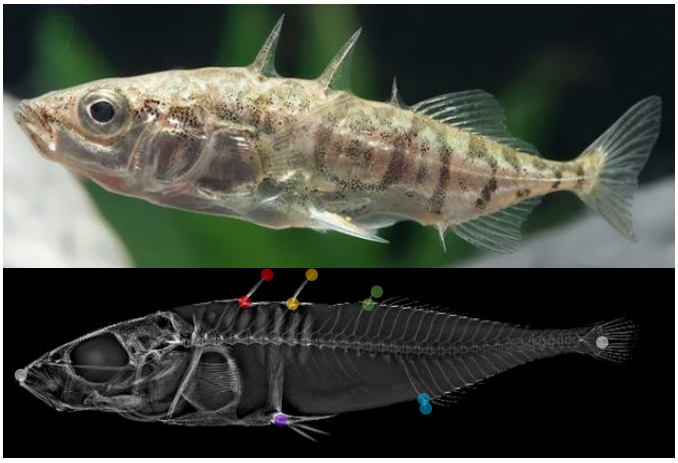


Figure 1 Threespine stickleback, alive (top) and X-rayed (bottom)
11 coordinates FishPose aims to predict are labeled on the X-ray image. Except for pelvis (purple), each pair marks the length of an element: White: standard length. Red, yellow and green: 1st, 2nd and 3rd dorsal spine. Blue: anal spine. ~4% fish have a 4th spine, which is labeled as the 12th and 13th coordinates when present.

Related works

Although intuitively, object detection algorithms[4] and X-ray image classification[5,6] seems relevant, the task is actually more directly related to pose estimation: given an image, the algorithm identifies the coordinates of certain key points, whether they be human joints or fish bone tips.

DeepPose was the first to tackle human pose estimation using deep learning[7]. The authors implemented a 7-layer convolutional neural net and trained the model to minimize the distances between ground-truth and predicted coordinates. Later deep-learning pose estimators achieve better and better metrics[8]. Recent advances have enabled real-time, multi-person pose estimation of ambiguous pictures[10,11].

Unlike the task these latest human pose estimators were addressing, FishPose has its unique advantages and challenges. Due to the scientific nature, our input images are unambiguous, with all parts of a fish clearly visible and well separated from other fish. Furthermore, fish poses are much more inflexible compared to that of humans. On the other hand, while regular human beings all have the same number of joints, fish

have variation in numbers of skeletal elements, even within the same species[12,13]. Thus, in addition to predicting where a mark is, FishPose also has to predict whether the mark exist at all. Furthermore, the amount of data is limited. The goal is therefore not to build a versatile model that can handle any weird X-ray images, but a simple model that performs well with limited training data.

Network implementation

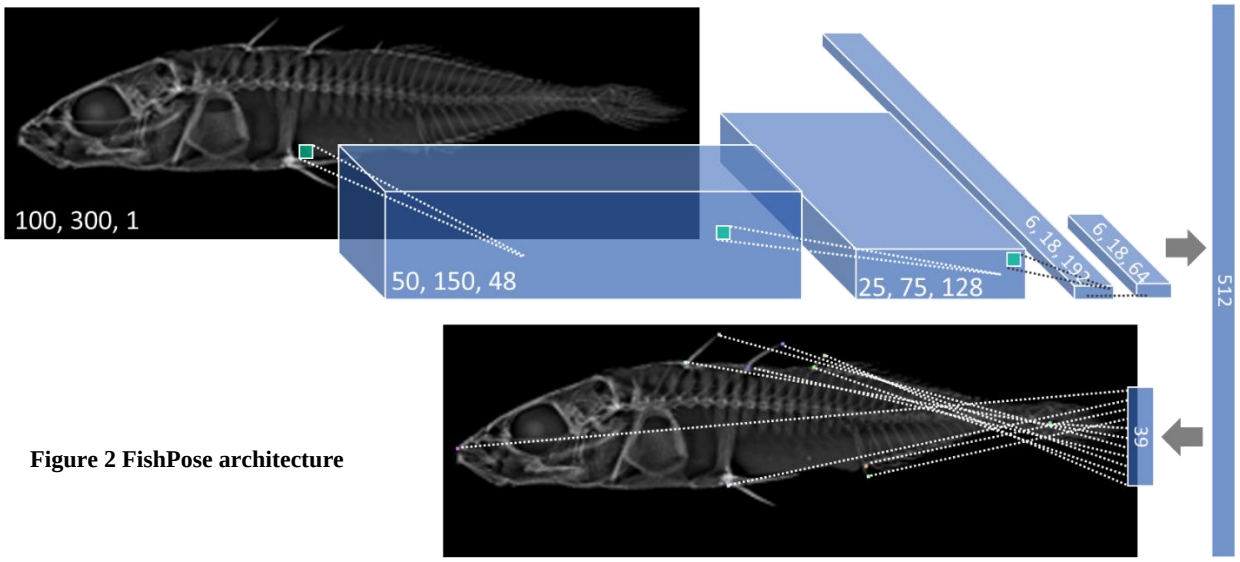


Figure 2 FishPose architecture

The model, adapted from DeepPose[7], is a six-layer neural network in the shapes *C1*(50 x 150 x 48) – ReLU – *P* – *C2*(25 x 75 x 128) – ReLU – *P* – *C3*(6 x 18 x 192) – ReLU – *C4*(6 x 18 x 64) – ReLU – *P* – *F*(512) – ReLU – *F*(39), where *C* denotes a convolutional layer, *P* a max-pooling layer with strides of 2 and *F* a fully connected layer. The filter sizes are all 5 x 5, except for *C4*, which uses a 1x 1 filter.

26 elements of the final layer denote the coordinates of the 13 marks (see Figure 1), while the remaining 13 denote whether they are present or not. L2 norm, which calculates the distances between predicted coordinates and the ground truth, is used to compute the "coordinate loss" when a mark is present. On the other hand, I use binary cross entropy to calculate the "presence loss". The final loss is the sum of coordinate loss and presence loss. To increase performance of coordinate precision, presence loss is capped at 0.05 (implemented as ReLU(presence_loss – 0.05)).

The loss is in turn minimized using Adam optimizer. Training is performed for 100 epochs with a minibatch size of 64 and a learning rate of 0.001 that reduces 6% for every 200 steps.

Dataset

The training set contained 3,332 fish and the validation set had 24 fish.

Our lab has 2,956 X-ray images that contains a total of 6,896 fishes. Of these fish, 931 threespine sticklebacks were labeled to some extent: 179 fish with only standard length labeled, 185 fish with 3rd dorsal spine and anal spine labeled, 55 fish with all but standard length labeled, and 512 fish with all five elements labeled. I further labeled 759 fish, which have the pelvis and occasional 4th spine additionally labeled, with a little help from my friends. The validation set was picked from these 759 fish. All other fish were augmented by flipping horizontally.

For input images, individual fishes were cropped out into 100 * 300 px boxes with proprietary non-deep learning code. The labels (coordinates of markers) were normalized to the box dimensions and center coordinate.

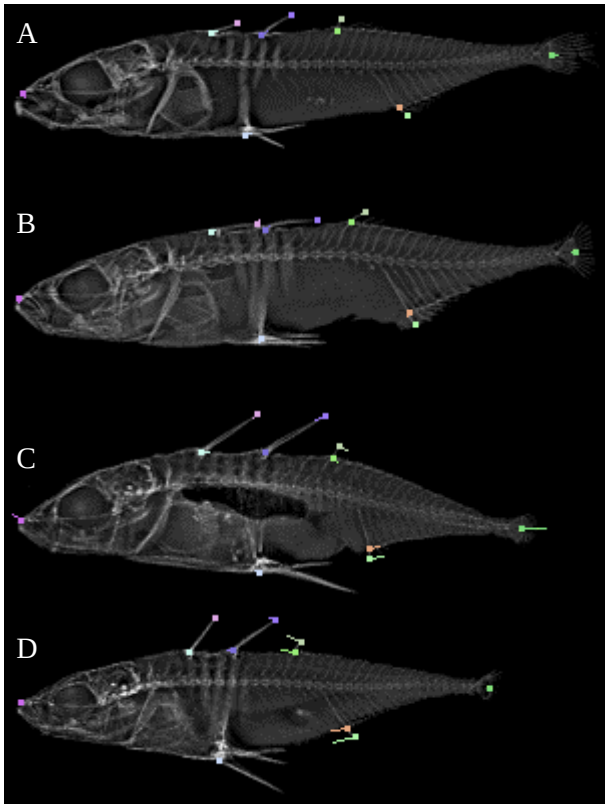


Figure 3 Representative predictions of validation images
Dots mark ground truths, and lines extend from ground truths to predicted coordinates.

Results

Figure 3 showed four predictions made by the baseline model. The model predicted even the folded spines in B correctly. C was a rare case where the head and tail coordinates had large deviation. Both C and D demonstrated that FishPose still had trouble identifying smaller spines.

Comparison with previous models

Percentage of correct key-points (PCK) was a common metric for assessing human pose estimators. It calculates how many predicted joints are within a certain threshold from their ground truth. A common threshold is 0.5 * head size[7,8]. Since I did not explicitly measure the fish head, I instead used 0.05 * fish standard length as the threshold. The baseline model achieved a PCK@0.05 of 0.9886, among the highest of pose estimators[8,9]. The high performance was of no surprise considering the relatively simple and invariable inputs.

Individual marks

As shown in **Error! Reference source not found.**A, the median performance of every mark fell within a few pixels from ground truth. However, larger deviations were still common, especially for tips of spines (*_t).

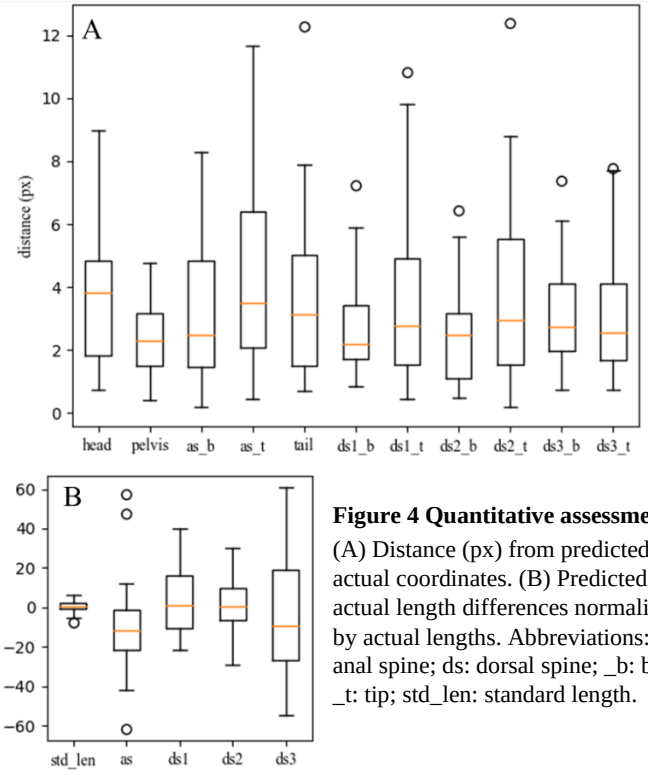


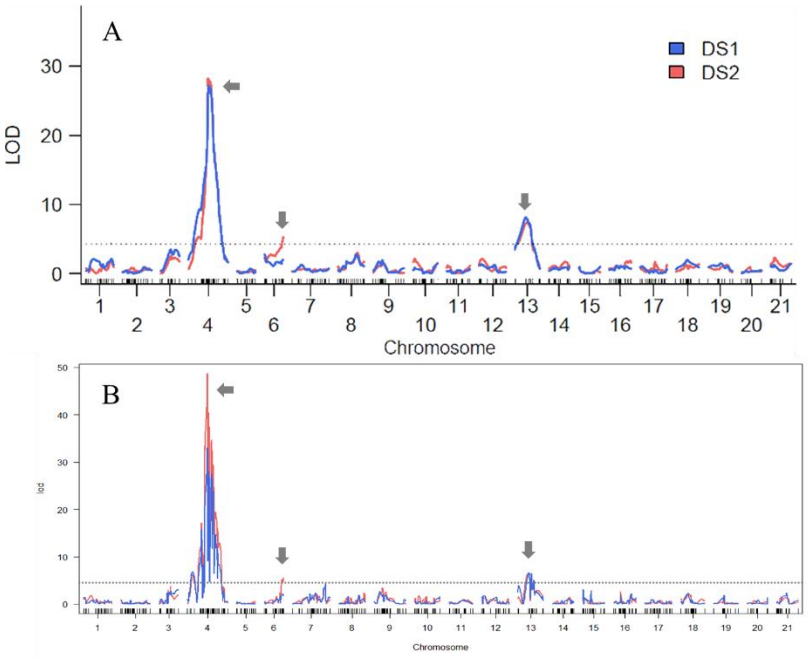
Figure 4 Quantitative assessments
(A) Distance (px) from predicted to actual coordinates. (B) Predicted – actual length differences normalized by actual lengths. Abbreviations: as: anal spine; ds: dorsal spine; _b: base; _t: tip; std_len: standard length.

Measuring lengths

The original purpose of FishPose was to measure lengths of bones of interest. Figure 4B quantified how well the model performed, which we can see was anything but well. Except for the standard length, predicted lengths easily deviated more than 10% from the truth. The performance discrepancy between individual mark and length measurements came from the fact that many spines were short, where a few pixels' deviation could result in major measuring errors.

Discovering genomic regions responsible for skeletal traits

Despite of that, FishPose is already useful. Figure 5A shows the result of quantitative trait loci (QTL) mapping by Howes *at al*[3]. The analysis required manually measuring various lengths of 590 fish, determining their genotypes on 466 different genomic positions, and examining which positions' genotype predicted certain phenotypes. I used FishPose to measure the lengths of the same fish (which was not in the training set nor the validation set), and followed the analysis[14]. Even though not all predictions were precise (Figure 5C), remarkably, all significant peaks were recovered, including the major peak on chromosome 4, the minor peak on chromosome 13, and the small peak at the end of chromosome 6 for dorsal spine 2 (Figure 5B, arrows). As pathetic the performance as shown in Figure 4B, FishPose is good enough for such genome-wide scale analysis, and is promising for discovering candidate genomic regions and phenotypes for further experiments.



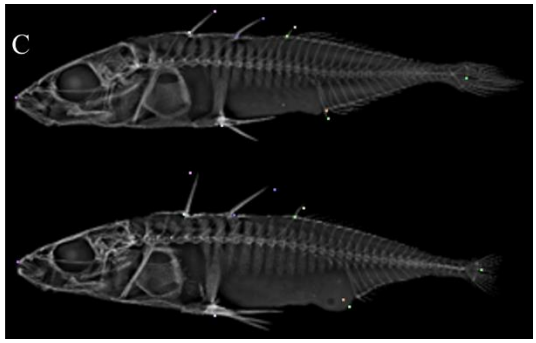


Figure 5 Genomic regions controlling lengths of dorsal spine 1 and 2

(A) Adapted from Howes *et al* (2017). (B) QTL analysis using lengths measured by FishPose. Dashed lines: significance threshold obtained through permutations tests ($\alpha = 0.05$). Tick marks on x-axis correspond to marker positions on the linkage map. (C) Two fish with marks predicted by FishPose. The prediction of the upper one is good, and that of the lower one is not.

Presence and absence of individual marks

FishPose was supposed to determine whether a mark was present. Unfortunately, due to unbalanced data, the model's predictions were invariable (always present or absent depending on the mark.) This is especially problematic for the rare (~3%) 4th spine, whose identification is of tremendous interest to the lab. Using weighted cross entropy to punish false negative did not help. The model never improved on recognizing the 4th spine until it started predicting the 4th spine at random locations.

Conclusion and Future Works

FishPose predicts the coordinates of 11 anatomical marks given a Xray image. With a simple 6-layer-ConvNet and a training set of 3,332 fish, the model performed reasonably well. Remarkably, with mediocre precision, FishPose recapitulated the result of a major QTL mapping. Such accomplishment demonstrated that days of labor in measuring length can be exempted with FishPose.

More data

The model approached subpixel precision for the training set, suggesting that it needs more training data to capture the full range of diversity in stickleback "poses". There are still more than two thousand unlabeled fish in the lab database. Incorporating all of them into training set would further this goal. Importantly, this would also increase the incidences of rare fish phenotypes (eg. 4-spine), creating a wider freedom for data augmentation. Experiences in recruiting friends and family to annotate suggested that the most efficient way to achieve this goal would be through training a few paid workers personally, regarding the amount of prior knowledge required for labeling.

Additional data augmentation

Horizontal flipping has enabled the model to handle both orientations well. Considering that the model had most trouble pinpointing the spine tips, artificially rotate the spines in the images can hopefully increase training set diversity and improve the model.

With a similar technique, adding or deleting spine can help balance the dataset, and help FishPose make meaningful predictions on the presence and absence of anatomical marks.

Transferability of the model

We have a smaller dataset of images of closely related fish species. By testing how well FishPose, with minor tweaks in architecture, performs on those data, I can examinet how easily applicable the model is to a broader research community, where people study not only skeletal evolution of various fish species, but also pigeons, snakes, lizards and mice as well.

Acknowledgements

Shubhang Desai provided useful insights as the course TA. Past and current members of the Kingsley lab collected the fish and Xray images. Julia Wucherpfennig, Garrett Kingman and Veronica Behrens provided guidance on the dataset. HsingHung Yeh, Rachel Grant and Heidi Chen gave useful advice on the slides that teach people how to label fish. David Lee made most of the pre-existing annotations. HuiYing Lu, my mother, labeled 200 fish for me.

Appendix: enhancing dorsal spine coordinate precision (in vain)

This section describes the myriad of futile attempts to help FishPose better predict the positions of dorsal spines.

Hourglass model

It was an unpleasant surprise that the model did not readily learn the features of the spine tips. This could be due to failure of propagating high-resolution information from lower layers to the final decision-making layer. To address this problem, I implemented a simplified hourglass architecture.

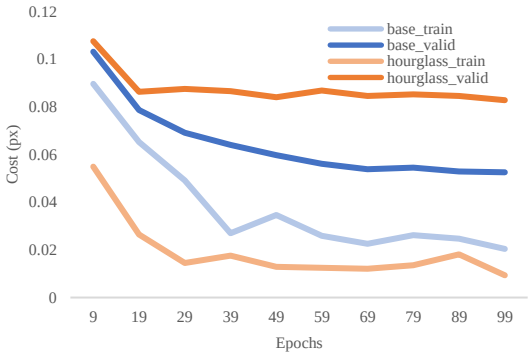


Figure 6 Cost during gradient descent

Compared with baseline model, the hourglass model overfitted the training set and performs poorly on the validation set.

Inspired by [9], the network had layers in the shapes *C1*(50 x 150 x 64) – ReLU – *P* – *C2*(12 x 37 x 128) – ReLU – *P* – *C3*(6 x 18 x 256) – ReLU – *P* – *C4*(3 x 9 x 256) – ReLU – *C5*(6 x 18 x 256) – ReLU – *C6*(12 x 37 x 128) – ReLU – *C7*(50 x 150 x 64) – ReLU – *F*(512) – ReLU – *F*(20). The filter sizes are 5 x 5 for *C1* and *C2*, 3 x 3 for *C3* and 1 x 1 for the rest. In addition, the activations of *C1*, *C2* and *C3* undergoes another 1 x 1 convolution and the values are added elementwise to *C7*, *C6* and *C5*, respectively.

Unfortunately, during training, while the training set cost continued to go down with each epoch, the validation set plateaued after only 20 epochs, and is worse than the baseline model (Figure 6). The increased number of parameters, compounded with small data size, likely made this more complicated model suffered from high variance problem. I tried skipping the 3rd and 5th layer to reduce complexity, but it did not help.

Object identification for spines

An alternative approach was to view all spines as equal (instead of 1st, 2nd …) and identify their positions with an object identification algorithm. I implemented a simplified YOLO model[4] and incorporated it into the baseline model. Specifically, an image was divided into 67 horizontal grids. The grid responsible for detecting a spine was the one containing its midpoint. The output of C3 of the baseline model was resized to (1,67), went through two 1 x 1 convolutional layers with 64 and 5 filters respectively. The final 5 filters indicated the presence/absence of a spine, and the xy coordinates of the two ends, respectively.

This model failed catastrophically. It could not identify most spines. The few predictions were not precise. Nevertheless, I do think this approach is promising for analyzing repetitive structures that varies in number (eg. spines, vertebra). To improve the model without an insurmountable increase in dataset volume, a probable fix is to have a different grid scheme. 67 horizontal grids were necessary for separating all spines. However, this meant that most grids would not have any spine, creating unbalance. While adding anchor boxes (the spines have similar aspect ratios) or adding vertical grids (spines have nearly identical y-coordinates) are useless, variable grid width, sampling more densely at mid-fish, could reduce grid number and possibly enhance model performance.

Data augmentation attempts

I tried cropping out the spine regions and added them into the training set, to give the model more practice on the spines. While the model did not perform better on the spines, it did worse on other marks.

Mid points

In order for the model to learn that the predicted line should be on the spine, I computationally asked the model to also predict the mid-points of the spines, which, unlike the tips, should fall on high pixel intensities. Performance was not enhanced significantly: tips of spines could still be in random places. Nevertheless,

the model did learn that the midpoint, the tip and the base should form a line. Good job for learning something useless to humans, model.

Alternative loss function

To come up with a way to punish every point along the predicted line that deviated from the spine, I implemented a third loss function to go along side "presence loss" and "coordinate loss": I asked the model to minimize the area between predicted and actual spines. It did not increase the performance.

The fundamental reason that data augmentation attempts, mid points and alternative loss function failed to improve model performance was that they helped by enhancing training, but the model was already impeccably good at the training set. It just had trouble extracting generalized rules and apply them on the validation set.

References

1. Thompson, A. C. *et al.* A novel enhancer near the Pitx1 gene influences development and evolution of pelvic appendages in vertebrates. *Elife* **7**, (2018).
2. Shapiro, M. D. *et al.* Genetic and developmental basis of evolutionary pelvic reduction in threespine sticklebacks. *Nature* **428**, 717–723 (2004).
3. Howes, T. R., Summers, B. R. & Kingsley, D. M. Dorsal spine evolution in threespine sticklebacks via a splicing change in MSX2A. *BMC Biol.* **15**, 115 (2017).
4. Redmon, J. & Farhadi, A. YOLOv3: An Incremental Improvement. (2018).
5. Al-antari, M. A., Al-masni, M. A., Choi, M.-T., Han, S.-M. & Kim, T.-S. A fully integrated computer-aided diagnosis system for digital X-ray mammograms via deep learning detection, segmentation, and classification. *Int. J. Med. Inform.* **117**, 44–54 (2018).
6. Gordienko, Y. *et al.* Deep Learning with Lung Segmentation and Bone Shadow Exclusion Techniques for Chest X-Ray Analysis of Lung Cancer BT - Advances in Computer Science for Engineering and Education. in (eds. Hu, Z., Petoukhov, S., Dychka, I. & He, M.) 638–647 (Springer International Publishing, 2019).
7. Toshev, A. & Szegedy, C. DeepPose: Human Pose Estimation via Deep Neural Networks. *2014 IEEE Conf. Comput. Vis. Pattern Recognit.* (2014). doi:10.1109/cvpr.2014.214
8. Chandra Babu, S. A 2019 guide to Human Pose Estimation with Deep Learning. (2019). Available at: https://nanonets.com/blog/human-pose-estimation-2d-guide/.
9. Newell, A., Yang, K. & Deng, J. Stacked Hourglass Networks for Human Pose Estimation. (2016).
10. Cao, Z., Hidalgo, G., Simon, T., Wei, S.-E. & Sheikh, Y. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. (2018).
11. Fang, H.-S., Xie, S., Tai, Y.-W. & Lu, C. RMPE: Regional Multi-person Pose Estimation. (2016).
12. Lindsey, C. C. 3 Factors Controlling Meristic Variation. in *The Physiology of Developing Fish* (eds. Hoar, W. S. & Randall, D. J. B. T.-F. P.) **11**, 197–274 (Academic Press, 1988).
13. Lindsey, C. C. Experimental study of meristic variation in a population of threespine sticklebacks, Gasterosteus aculeatus. *Can. J. Zool.* **40**, 271–312 (1962).
14. Broman, K. W., Wu, H., Sen, S. & Churchill, G. A. R/qtl: QTL mapping in experimental crosses.

Bioinformatics **19**, 889–890 (2003).