# GoFigure and The Digital Fish Project: Open tools and open data for an imaging based approach to systems biology

*Release 0.91*

Alexandre Gouaillard, Titus Brown, Marianne Bronner-Fraser, Scott E. Fraser, and Sean G. Megason

September 11, 2007

Center of Excellence in Genomic Science and Division of Biology, California Institute of Technology, Pasadena, CA 91125

**Abstract**

As part of the Center of Excellence in Genomic Science at Caltech, we have initiated the Digital Fish Project. Our goal is to use *in toto* imaging of developing transgenic zebrafish embryos on a genomic scale to acquire digital, quantitative, cell-based, molecular data suitable for modelling the biological circuits that turn an egg into an embryo. *In toto* imaging uses confocal/2-photon microscopy to capture the entire volume of organs and eventually whole embryos at cellular resolution every few minutes in living specimens thoughout their development. The embryos are labelled such that nuclei are one color and cell membranes another color to allow cells to be segmented and tracked as they move and divide. The use of a transgenic marker in a third color allows a variety of molecular data to be marked. *In toto* imaging generates 4-d image sets (xyzt) which can contain 100,000 to 1,000,000 images per experiment. We are developing a software package called GoFigure to visualize, segment, and analyze these very large image sets. GoFigure uses a MySQL database back end for managing storage of images and segmented objects and uses VTK and ITK for visualization and segmentation. We plan to use *in toto* imaging to digitize the complete expression and subcellular localization patterns of thousands of proteins throughout zebrafish embryogenesis. This genomic data, our zebrafish lines, and GoFigure will be distributed following the Open Data/Open Source model.

## Contents

## 1 Introduction

### 1.1 Systems Biology, Imaging, and Image Analysis

With the completion of the genome projects, the stage is now set for a more complete understanding of how animals are created. The genome projects have given us the complete sequence of DNA that encodes the blueprint of life for many animals including human, mouse, zebrafish, and fruit fly[2][3][4][5]. The challenge is now understanding how this code functions. The problem however is that the only thing that we can currently reliably predict from the genomic code is which parts can be transcribed and translated into protein. We cannot accurately predict when and where a protein will be expressed, we cannot predict how a protein will fold and function once it is translated, and we cannot predict the interaction between expressed proteins that allow them to form functional genetic circuits. Thus, although the genome projects have given us the complete code for constructing many organisms, the only part of the code we can currently read is the parts list. The challenge ahead is understanding how all these parts fit together to form molecular circuits, how these circuits process information to regulate the behavior of cells, and how the behavior of cells is orchestrated to generate functional form as in embryogenesis. This post-genomic endeavor has come to be called systems biology.

Although systems biology grew out of the “-omics” field which made heavy use of in vitro biochemical approaches (e.g. sequencing, microarrays, and proteomics), in vivo imaging is becoming an increasingly powerful tool for systems biology. In vivo imaging is advantageous over biochemical approaches for doing systems biology because of 4 considerations about how biological circuits function[1]. First, biological circuits function at the single cell level. Microscopic imaging easily achieves this resolution while in vitro techniques typically do not. Second, biological circuits function over time. With imaging, different components of a biological circuit can be labeled using different colors of fluorescent proteins and the dynamics of the circuit monitored non-invasively with time-lapse fluorescent imaging as the circuit functions in an intact system. In vitro biochemical approaches typically require the tissue to be destroyed in order to be assayed which precludes longitudinal analysis. Thirdly, the quantitative amounts of components in a biological circuit are important for its function. Fluorescent imaging can accurately quantitate the levels of molecular components even at the protein level through the use of fluorescent protein (e.g. GFP) fusions. Omic approaches are typically less quantitative and focus at the DNA or RNA level which is less relevant.

And finally, the anatomical context of biological circuits is essential for determining their function; the use of spatial cues to generate different cell types in different places is a fundamental aspect of development. In vivo imaging can capture data from intact animals preserving its anatomical context. In vitro approaches typically grind up the tissue to assay it, thus destroying its anatomy.

As you will see reading this paper, the use of imaging for systems biology opens up a number challenges in image analysis. Some of these challenges such as visualization of large image sets have close parallels in other areas such as medical imaging. Other challenges such as segmenting closely packed cells in space, across time, and across cell division can benefit from techniques borrowed from other areas but have unique differences requiring novel approaches. We believe that imaging based systems biology provides a novel growth opportunity for the image analysis field, and likewise that the sucessful development of image analysis tools is absolutely essential for the progress of systems biology

### 1.2 CEGS and the Digital Fish Project

We recently received a grant from the National Human Genome Research Institute to establish a Center of Excellence in Genomic Science (CEGS) at Caltech. The CEGS program was established around 5 years ago to support high-risk/high-reward research with the potential for making a substantial impact on genomics. Around 1 CEGS has been funded each year since its inception. The focus of research of each CEGS is investigator led but taken together they span the range of genomics, nanotechnology, and systems biology. Our CEGS is more focussed on imaging, systems biology, and embryogenesis relative to the others and is titled “*In toto* genomic analysis of vertebrate development”. Our CEGS has 4 specific aims: 1) To develop *in toto* imaging to allow us to image and track every cell in developing embryos and digitize fluorescently marked data at the level of the cell; 2) develop our FlipTrap technology to allow us to generate transgenic animals that fluorescently mark a range of biological data including protein expression and subcellular localization as well as mutant phenotype; 3) develop our Hybridization Chain Reaction (HCR) technology to detect a wide range of biological substrates including RNA, protein, and metabolites; and 4) to integrate and scale-up the above aims to create a Digital Fish by using in toto imaging to digitize expression patterns and mutant phenotypes at cellular reolution in living embryos on a genomic scale and use this data to construct computer models of developmental processes.

Zebrafish is the main model system we are employing with a lesser focus on quail. Zebrafish (*Danio rerio*) is a small, freshwater, tropical fish commonly available in pet stores. It has become a popular model system for the study of genetics and development in the last 2 decades and there are now many labs worldwide that study it. There have been several large-scale mutagenesis screens caried out in zebrafish and its genome has been largely sequenced. Zebrafish has the advantages of fruit fly in that it is amenable to forward genetic screens, but unlike fruit fly, zebrafish is a vertebrate so is much more relevent to humans. Another huge advantage of zebrafish is its suitability for imaging. Zebrafish embryos are transparent, small, develop freely outside their mother, and develop directly from egg to adult without any larval stages. This means that a zebrafish egg can be placed under a microscope and continuously imaged throughout embryogenesis. This would be impossible with a mouse which develop inside its mother, with a frog (Xenopus) egg which is opaque, or with a fruit fly (Drosophila) which has several larval/pupal stages and is opaque.

## 2 *In toto* imaging

The goal of *in toto* imaging is to image and track every single cell in a developing tissue, organ, or eventually whole embryo[6]. There are several steps to *in toto* imaging. The embryos must first be labeled with "segmentation markers" which allow all the cells to be segmented. Additionally embryos can be labelled with additional markers to reveal RNA or protein expression. The next step is to acquire xyzt image sets using confocal or 2-photon fluorescent microscopy. And the final step is processing the often very large image sets to segment out all the cell trajectories and to extract quantitative, cell-centric data. This final image analysis step is the main focus of this paper and the goal of GoFigure.

## 2.1 Embryo labeling

Since we use fluorescence microscopy, the embryos must be labeled with fluorescent markers and since we are doing time-lapse imaging of living embryos, we must used vital labels. Fluorescent proteins such as GFP (Green Fluorescent Protein) are the perfect choice. Fluorescent proteins (FPs) are proteins cloned from a number of marine invertebrates including jellyfish and coral that have the unique property of generating a signal (fluorescence) without the addition of any exogenous substrate. The proteins can be expressed in a wide variety of animals and will fluoresce light after being excited with light of a shorter wavelength. There are now FPs available across the visible spectra including blue, cyan, green, yellow, orange, red, and far red. Since FPs are proteins rather than organically synthesized small molecule dyes, FPs can be endogenously expressed by transgenic organisms and are open to all the power of genetic engineering.

There are 2 basic uses of markers for *in toto* imaging: segmentation and expression. Segmentation markers allow the cells to be segmented in space, across time, and across cell-division. We typically use a histone-FP of one color (e.g. histone-cerulean) and a membrane-localized FP of another color (e.g. membrane-mCherry). This combination would generate embryos in which every nuclei is cyan and all the cell membranes are red. Histone-FP is a fusion protein between an FP and histone2B one of the core proteins that DNA is wound around to form chromatin. This marker is nice because it not only marks the nuclei during interphase but also during mitosis. Many other nuclear markers will become diffuse during cell divsion, but since the histone-FP is stuck to the DNA, it marks the condensed chromosomes of both daughter nuclei thoughout mitosis allowing the mother cell to be linked to its 2 daughters.

We key on nuclei rather than whole cells for the initial segmentation because nuclei have a simpler and more uniform shape than whole cells. Nuclei generally have some kind of ovoid shape (ball, coin, sausage) whereas whole cells can have a wide diversity of shape including spheres, columns, spindles, stars, or in the extreme case of neurons highly branched trees. The histone-FP marker alone can be used for nuclear based segmentation. This can work well on cultured cells, but in tissue of intact animals the nuclei are often very close together and appear to be touching in our fluorescent micrographs. For this reason, we use a second color to label the cell membranes. By subtracting the membrane channel from the nuclear channel, neighboring nuclei can be made more distinct and easier to segment. The membrane label has the aditional advantage that it defines the extent and shape of the whole cell. Cell morphology can be very useful for determining cell type and tissue type in fluorescent micrographs. The membrane marker also provides the opportunity to segment out the whole cell. For exampe, once all the nuclei have been segmented, regions can be grown outward until they hit the cell membrane marker to segment whole cells. If this dual segmentation is done for all cells, then it can be used to define the basic compartments of a cell: cell membrane, cytoplasm, and nucleus. This informations is very useful for cellular approaches because it defines the cellular geometry as well as molecular approaches since it can be used to define the subcellular localization of proteins.

In addition to segmentation markers, the other use of markers for *in toto* imaging is expression markers. This is a bit of a catch all term but refers to whatever else is being marked such that it can be digitized with *in toto* imaging. For example, transgenic fish can be made that express a fluorescent protein wherever some gene is normally expressed during development. Once all the cells have been segmented using the segmentation markers, the amount of the expression marker in each cell can then be quantitated at cellular resolution.

Our FlipTrapping approach is particularly useful in this regard because it generates fusions of endogenously produced proteins to YFP (yellow fluorescent protein). Since the YFP fusion protein is expressed from the same genetic locus as the endogenous protein, it should be expressed at the same levels and in the same tissues as the endogenous protein, and thus in toto imaging can be used to quantitate protein expression on a cell by cell level. FlipTrap fusion proteins also reveal the subcellular localization of the trapped protein.

## 2.2 Image acquisition

In toto imaging tracks cells during development only by the correspondence of cells from frame to frame. This is quite different than traditional fate mapping in developmental biology where only a small subset of cells is labeled at one stage of development and the location of their progeny is observed at a later stage. In toto imaging thus requires images that have high enough resolution in space to resolve all the cells (we typically aim for 1um resolution) and high enough resolution in time to not lose track of moving and dividing cells (this requires around 2 minute time resolution). In addition to high resolution, in toto imaging also requires complete coverage. All the cells of interest must be continuously imaged in order to be tracked; if cells leave the field of view then their lineage cannot be completely tracked.

Achieving both high spatial and temporal resolution as well as complete coverage can be achieved through the use of time-lapse confocal or 2-photon microscopy. Confocal microscopy uses a pinhole to eliminate out of focus light whereas 2-photon microscopy only excites fluorescence in the focal plane. Thus both techniques allow thin (˜1 um) optical sections of fluorescent samples to be captured. By imaging at a series of focal planes, stacks of optical sections can be captured to generate a volumetric image and this can be repeated over time to generate xyzt images. The axial (z) resolution with these techniques is typically around 2-fold less than the planar (xy) resolution so the volumetric images are anisotropic. Confocal imaging provides 2-fold higer resolution than 2-photon imaging but has a depth penetration limit of ˜150 um. 2-photon imaging can image all the way through a zebrafish embryo and has reduced phototoxicity to the embryo and photobleaching of the fluorescent labels ( see image 1 ).

For in toto imaging, we mount zebrafish embryos in custom developed chambers that hold the embryos stationary for imaging while allowing them to grow and develop normally. We mount the embryos shortly after fertilization and can continuosly image them throughout embryogenesis (3 days). The use of a motorized stage permits tiling xyzt images across a single embryo and imaging multiple embryos.

## 2.3 Image processing

After the embryos have been labeled and imaged, the final challenge of in toto imaging is image processing. The goal of image processing is to track all the cell movements and divisions to generate cell lineage trees, to define the boundaries of all cells and their compartments (nucleus, cytoplasm, membrane, extracellular space), and to quantitate the level of fluorescence within each cell and subcellular compartment. We are developing a software package called GoFigure for handling all of the image processing needs of in toto imaging so this step will be further elaborated in the next section.

## 2.4 Significance

There are a number of significant outcomes that we hope to acheive with the Digital Fish Project using in toto imaging. The first is to obtain a complete understanding of the cellular basis of development through the construction of complete lineage trees. We plan to use in toto imaging to image and digitize all the cell

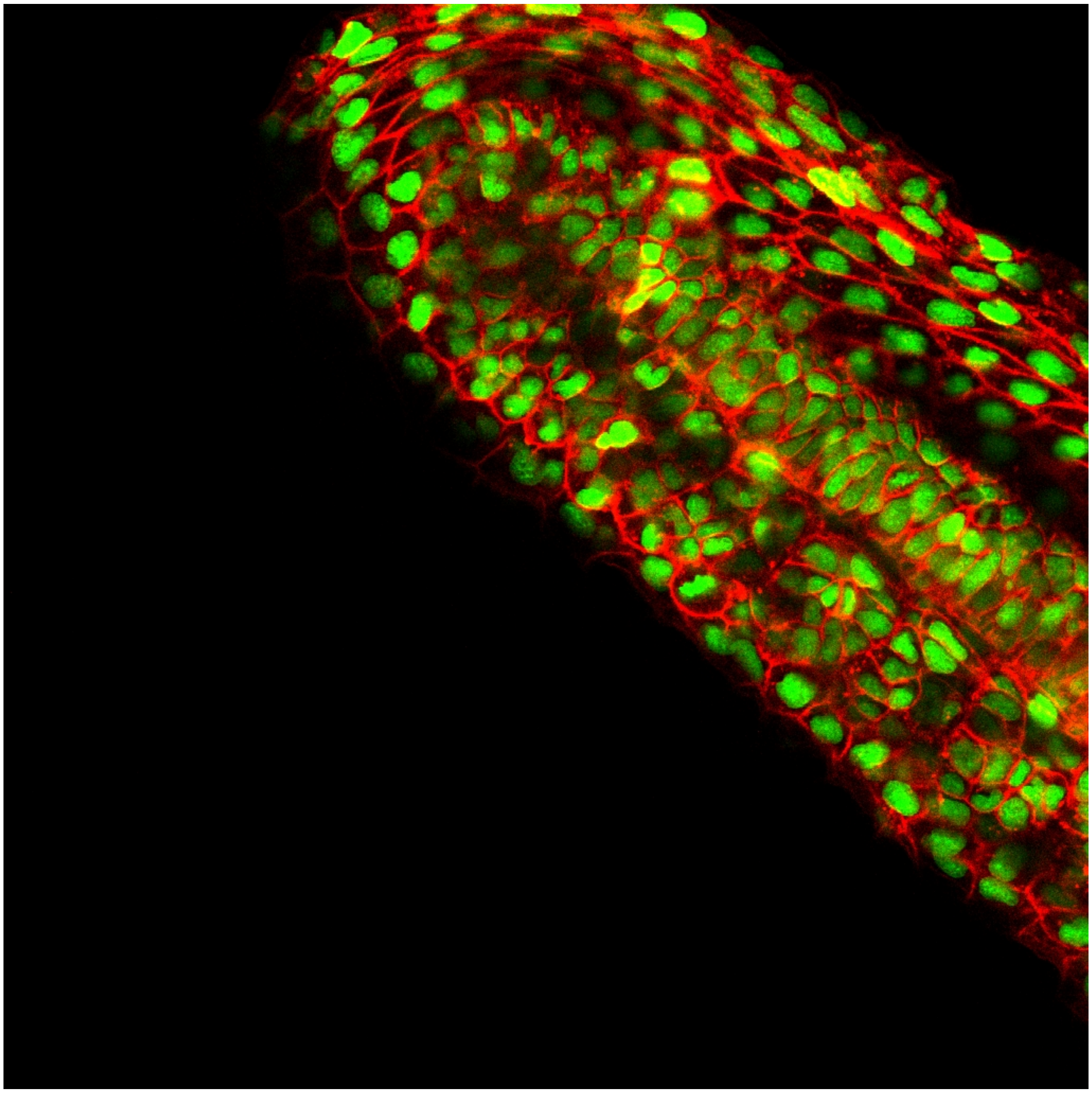

Figure 1: This represents the tailbud of the zebrafish embryo. The image has been taken using a 2-poton confocal microscope.

movements, cell divisions, cell deaths, and cell shape changes that are used to construct organs such as the inner ear, eye, and spinal cord and we hope to eventually apply this approach to the whole embryo beginning with the fertilized egg. This type of approach will give us a complete understanding of morphogenesis at a cellular/morphological level (but not molecular).

A second use of in toto imaging is to digitize molecular data for use in systems biology. Transgenic fish such as FlipTraps can be in toto imaged and the segmentation masks used to quantitatively define the fluorescence in each cell and subcellular compartment over time. Since FlipTraps generate fluorescent fusion proteins, the levels of fluorescence directly correlate with the number of molecules of the tagged protein. Thus with the proper controls and corrections, in toto imaging can be used to get accurate measurements of the number of molecules of a protein in all cells during a developmental process. Such data is extremely useful for doing systems biology. For example, by using a few colors to mark the critical nodes of a biological circuit, it is possible to monitor the function of that circuit dynamically in live embryos[1]. In toto imaging can also be used on a large scale in a high-throughput fashion to digitize molecular data. We hope to use such an approach to capture near single cell resolution, 4-d representations of protein expression data using our FlipTraps. This data will be integrated to form the Digital Fish.

## 3 GoFigure

### 3.1 Goal

GoFigure is a software application we are developing to serve as the principle image analysis tool for in toto imaging. The goal of GoFigure is to digitize image based data at the cellular level. Confocal/2-photon microscopy generates image sets comprised of pixels but it must be translated into cells for data analysis for biologists. The cell is the basic building block of embryos as well as a fundamental unit of computation in biological circuits since most molecular components cannot pass the cell membrane. While humans have an easy time spotting cells in microscopic images, computers have a notoriously difficult time. One of the most significant challenges of GoFigure is cell segmentation. The datasets can be several terabytes in size which pauses computational issues. It is 3D+t datasets, whose visualization can be challenging. It can also be very noisy and/or presents (interlacing) acquisition artifacts. Once the cells are segmented in space, time, and once cell divisions have been taken into account, , it is straightforward to visualize the data and quantify the data. We intend GoFigure to be an integrated interface for segmentation, visualization, and cell-based analysis. It should be an user-friendly tool for biologists to use in their research. Although GoFigure is being created to serve the needs of in toto imaging and the Digital Fish Project, we hope that this software will also be used by other scientists doing imaging-based systems biology.

### 3.2 Technologies

The curent version is a windows-only research prototype. Coded entirely in C++, it uses high-level OO design, design patterns and other recent component oriented theories to assemble the 100+ classes into a large, but flexible application.

it is based on Microsoft Foundation Classes (MFC) for the Graphical User Interface (GUI) and database accesses. It is using a MySQL database for segmentation results and cell information storage. Links to datasets are stored in the database, but the datasets are actually stored outside of the database. VTK is used for data visualization.

## 3.3 Client-server architecture

GoFigure uses a client server architecture. The server hosts a MySql database which stores all of the segmentation results and cell analysis. Links to the images are stored, but the image themselves are kept on a different location. The client is the GoFigure software itself. GoFigure interacts with the (possibly remote) server during image analysis. Typically, GoFigure will load data from the server, allow analysis, processing and visualization of the data, and then store any new data in the database.

the client server architecture has several advantages. Most importantly, the database server allows very large data sets to be worked on. The MySql database server can manage data sets many terabytes in size. It also stores the data as it is created in the possibility the client crashes. The client-server architecture also allows several people to interact with the same server and dataset using different clients.

The server and the client can reside on different computers or can be located on one computer. For most applications, all of the components required for GoFigure can simply be installed on one computer which can serve as both client and server.

## 3.4 GUI

The GUI is based on MFC classes that give GoFigure the qualities of a professional applications. Based on the MDI design, it is composed of a main window that allows standard processes (loading new files, quiting the application, ...) while no child window is open. When opening a new document (i.e. a dataset stored in a database) a child window will appear, enabling menu bars and options depending on which data are available and / or visible.

An important concept in GoFigure are the “views”. Views are subwindows within the MDI child window that are dock-able and tab adjustable. GoFigure has quite a few of these available: XYZ or XYT composite viewers, slice informations viewer, dataset slice / file viewer and database table viewer. The end user can define the layout of the views according to his own preference or the experiment he is working on. GoFigure stores the curent layout in the registry and thus, “remembers” the layout across sessions.

As most people working on real cases know, there is no such thing as an overall general segmentation algorithm. Whatever algorithm we choose, it will never give perfect results. It is thus important to add user interaction to the segmentation aspect of GoFigure. It will not only impact the GUI layer, but also the segmentation layer. We will only adress the GUI layer here.

1. GoFigure allows you to interactively draw a Region Of Interest (ROI) that will be used as a constraint by the segmentation algorithms. If you define the ROI to be an anatomical feature, that can lead to tracing the partial lineage for that specific features (the ear, for example)

2. It is possible to run the algorithm on a single cell instead of running it on an entire image or on all a ROI. In that case a slightly different algorithm is used. It allows one to complete the results by manually targeting the cells that the global algorithm would have missed ( false negatives ). Aiming for the lineage, we need to insure that all cells are extracted.

3. GoFigure also allows you, after segmentation, to individually pick any result (any cell) and remove it, thus dealing wth the false positives. You can either pick the results in the Table viewer ,which in turn will colored the representation of the corresponding cell on the XYZ / XYT viewer, or vice-versa, as all the views are linked.

## 3.5 Figures, meshes, tracks, and lineages

GoFigure defines and handle four kinds of objects: figures, meshes, tacks and lineages.

Figures are the 2D contours of the nucleus of a cell within a 2D XY slice of a dataset, at a given depth (Z) and time (T).

Meshes are the 3D surfaces of the nucleus of a cell within a 3D XY-Z stack of 2D XY slices for all depths (Z) at a given time. T. It can also be seen as a collection of figures, and in GoFigure, all the figures have a link to their corresponding mesh. Depending on the segmentation algorithm, we can start from either figures or meshes.

Tracks are the continuation of meshes in time. You can display it as a “tube” representing all the positions of a given mesh during a certain amount of time. Accordingly, all meshes in GoFigure have a link to their corresponding track.

Lineages are the goal of the segmentation process. It is composed of the addition of all tracks, plus their branching (where cells subdivide). Again, each track in GoFigure has a link to its lineage. Note that lineages are, by definition binary trees.

## 3.6 Image segmentation

Currently, we use a segmentation algorithm to obtain the figures on a dataset, and we then group them into meshes, tracks, and lineage using a propagation algorithm that searches the local neighborhood (with respect to the dimension of the space) for the next object. For example, once the figures are available, we search within the Z stack for other figures belonging to the same mesh. Similarly, given the meshes, we look across time for its next and previous positions.

The current segmentation algorithm for figures is quite accurate (85 percents or more of the cells are found) and has the advantage of directly extracting all the biological values of interest, but unfortunately is too slow (more than a minute per 2D image) to accommodate with the size of the dataset we are expecting. We are experimenting with other algorithms, possibly working directly at the 3D, mesh, level.

The sampling in time is good enough for simple neighborhood search algorithm to be used. This supposes that the figure segmentation is complete and accurate to begin with, since errors are cumulative. We are working on tracking algortihms, possible guided by a motion estimation of the cells, that can enhance the results of a previous segmentation. The mesh segmentation algorithm could for example expect the cell to exist across several slices, and if a figure is missing between two existing related, figures, the algorithm can fill in the gap.

## 3.7 Visualization

All the visualization is based on VTK. We, at most, visualize a Z-stack and a T-stack for a given common XY slice at a time. We do not really suffer, as long as visualization is concerned, from the size of the dataset. Moreover, the depth and resolution in Z are usually limited in confocal imaging, thus the amount of memory needed for vizualization depends only on the time range and sampling.

The two main data visualization “views”, respectively XYZ and XYT viewers, are made of 4 render windows which are resizable. One window is used for visualizing the curent XY slice, two side windows are used for the projection on Z or T, and the last windows is used for a 3D volumetric representation of the curent Z/T stack. All 4 windows are linked, and any modification affects them all.

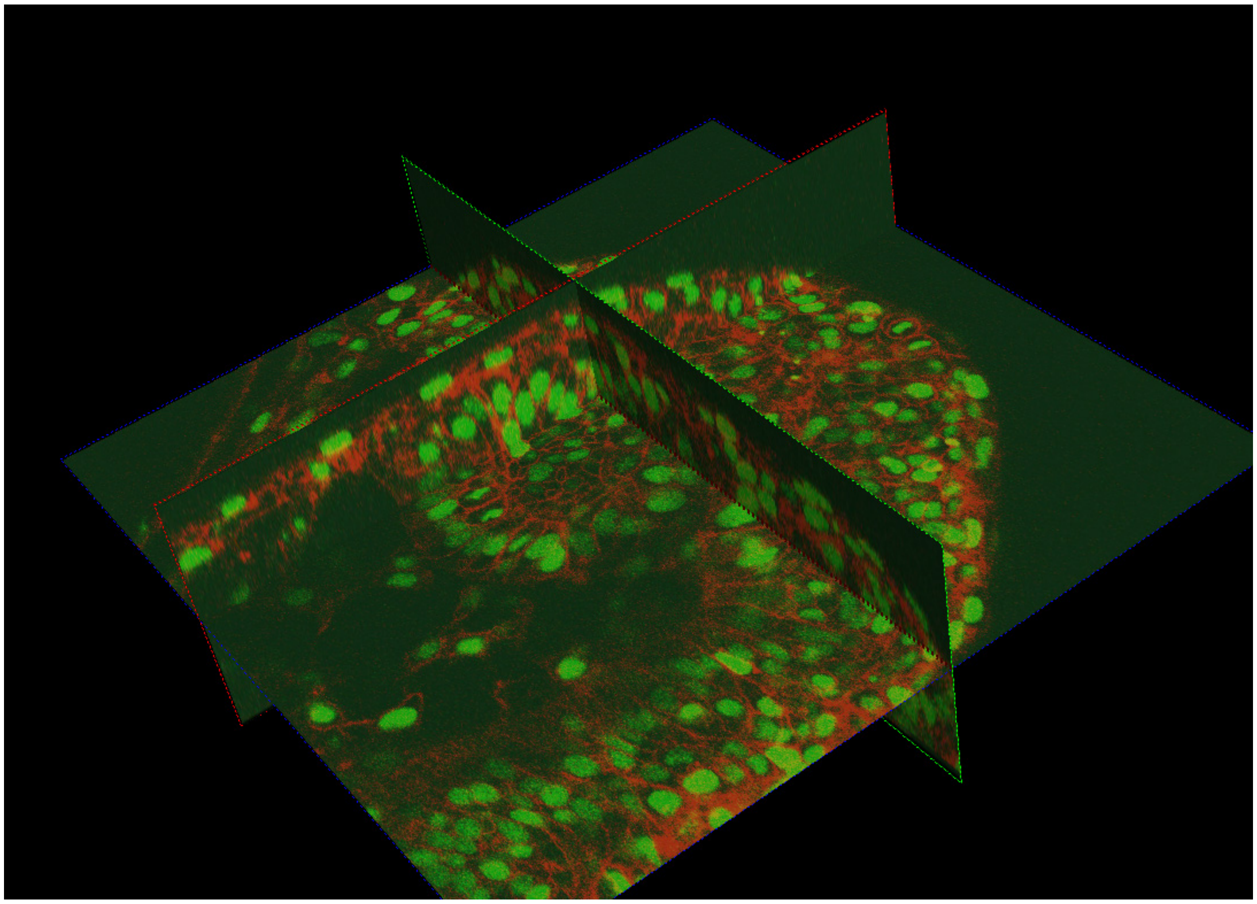

Figure 2: One of the four rendering wndows composing our XYZ view. Each plane is linked with the three other representations ( see **??** ). If the user modify the position of one plane, the four renderer are updated.

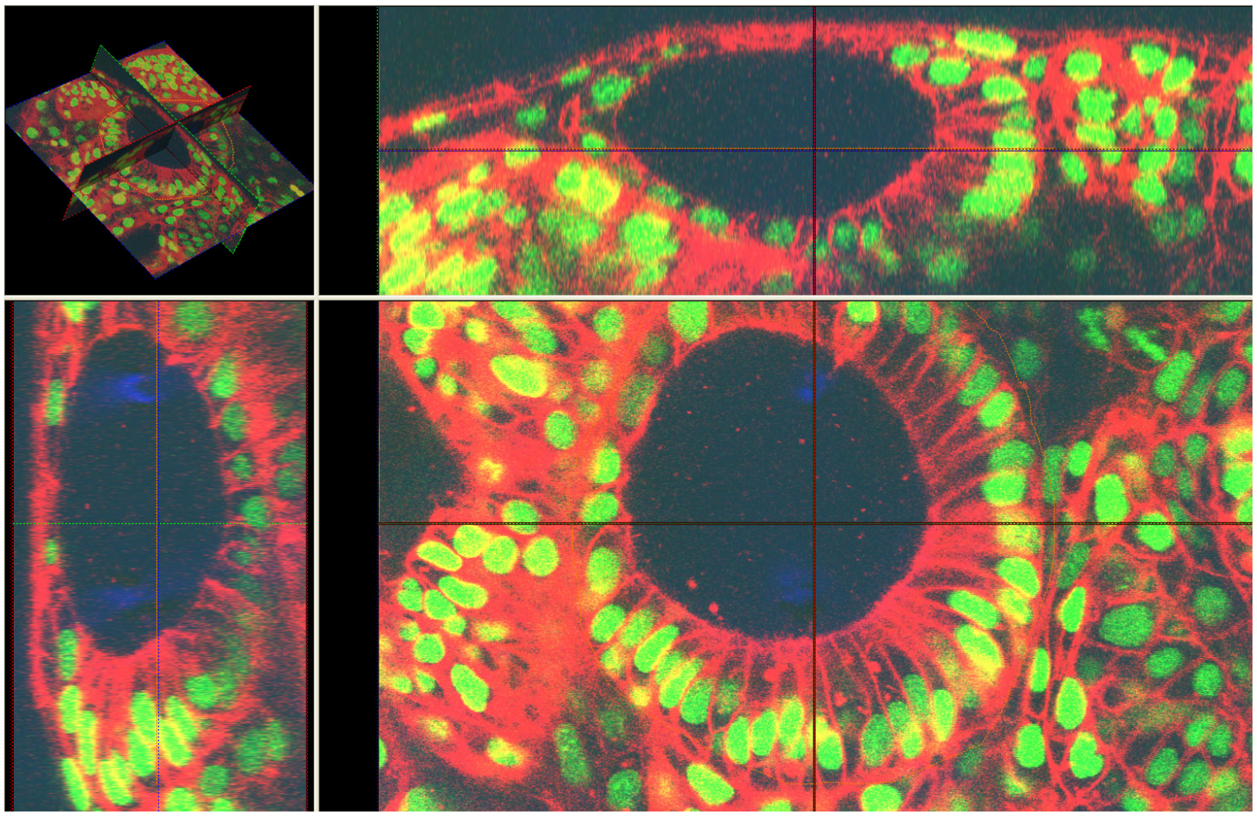
Figure 3: Our XYZViewer.

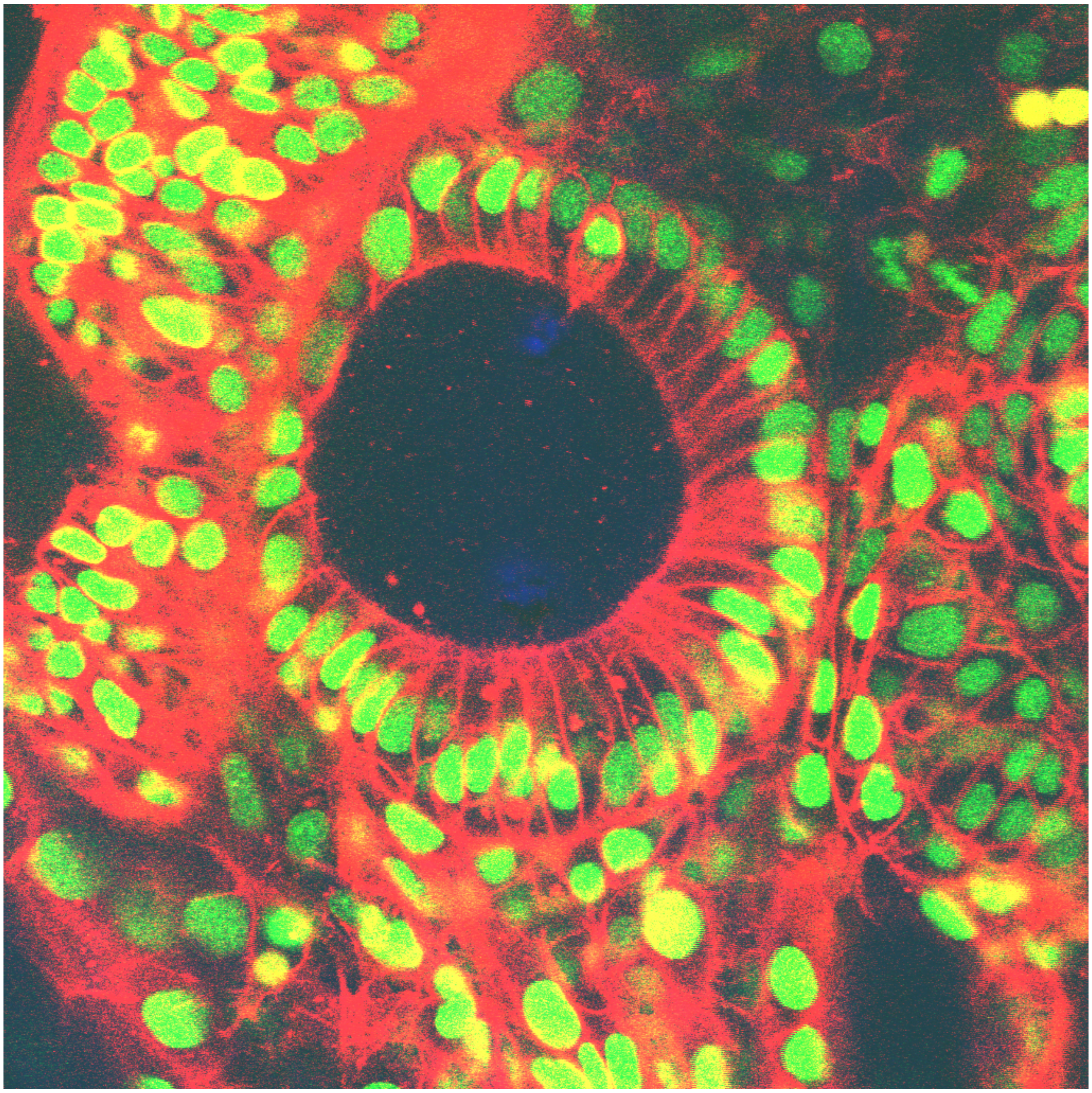

Figure 4: A high resolution zoom on the ear area of the zebrafish embryo (unfortunatly, the image had to be scaled down for the paper).

The result of the segmentations are added to the views as discrete meshes whose dimension depends on the type of object. We use the vtkPolyData structure for that.

### 3.8 Plans for future development

This paper is more a work in progress report than a technical report. We are willing to announce to the community that an open-source, open-data project is starting so that interested individuals or research group cam follow or even join the effort. As the software is based on NAMIC toolkits we thought that the Insight Journal, and MICCAI Open-source workshop was the best place to make such an announcement.

There is still a lot of work on the segmentation part and we are looking forward having motivated individuals join the development effort to work on that specific subject. As the project is Open Source, we are also hoping that other researchers from other institution will join the effort and help us develop these tools.

To ease the transfer, configuration and installation of GoFigure, we are planning to move from a windows-only code to a cross-platform code. Right now, we have already made one step toward it by moving the configuration process to CMake, one of the NA-MIC tool. CMake can handle cross-plateform configuration. CMake also comes with CPack, that we are using right now to create packages for GoFigure. We have set up a dashboard (Using Dart 2) to continuously build and monitor our code on different flavors of windows, using different versions and/or parameters of MSVC++. As for the compilation, to be truly cross-plateform, we need to replace MFC classes.

We already use VTK for visualization, CMake for configuration, and CPack for packaging. All those are part of the NA-MIC Toolkits. We are planing to replace the MFC-based GUI we have right now by a KWStyle GUI, and we are looking within the other toolkits for a replacement for the MFC-based database access management.

We are planning to improve the quality of our code by developping a test suite and benchmarks, and by monitoring the code coverage, the memory leaks. All those things will be directly available to us once we will have cross platform code with a test suite thanks to hte NA-MIC Tollkits integration ( CMake / CTest / Dart 2 / gcov / valgrind ).

The images we are acquiring are tiled, and so are our segmentation algorithms. We are planning to developp multithreaded versions of our algorithms to take advantage of the multi-core processors or to be run on clusters. We are planning to develop command line version of each feature of our code to be able to run them in batch mode.

## 4 Conclusions

As part of the CEGS based Digital Fish Project we intend to scale up our FlipTrap screen to generate ¿1000 transgenic lines of fish and to use in toto imaging to to digitize their 4d protein expression patterns. We will register these 4d data sets onto a common framework to form the Digital Fish, a vast anatomically-based warehouse of molecular data. This data will be made publically accesible via the web as it becomes available in accordance witht the Bermuda Principles which have been successful for genome sequence data. We also intend the Digital Fish to be more than just a data warehouse, but to form a framework for constructing cell-based computer models of developmental processes such that we can move towards the ability to actually compute phenotype from the code of genotype.

## 5 Links

Both Data and code are available under Creative Commons by attribution licence.

Data are available at the following link. More dataset will be added later.
http://www.insight-journal.org/midas/view_collection.php?collectionid=37
http://www.insight-journal.org/dspace/handle/123456789/580

The Research Prototype is available here:
http://gofigure.caltech.edu/beta/

The current work is documented here:
http://iorich.caltech.edu/gofigure/

The current installation process is documented here:
http://iorich.caltech.edu/gofigure/wiki/Installation

Access to the svn repository is documented here:
http://iorich.caltech.edu/gofigure/wiki/CodeVersioning

The doxygen-generated documentation of the source code is available here:
http://iorich.caltech.edu/gofigure-doc/doxygen/html/classes.html

The dart2 dashboard is visible here:
http://cegs-dart-server.caltech.edu/GOFIGURECore/Dashboard/

## References

[1] Megason SG and Fraser SE. (2007). "Watching biological circuits function: The role of imaging in systems biology", Cell, in press 1.1, 2.4

[2] Venter JC et al. (2001). The sequence of the human genome. Science. 291, 1304-51 . 1.1

[3] Lander ES et al. (2001). Initial sequencing and analysis of the human genome. Nature 409, 860-921. 1.1

[4] Waterston RH et al. (2002). Initial sequencing and comparative analysis of the mouse genome. Nature 420, 520-62. 1.1

[5] Jaillon O et al. (2004). Genome duplication in the teleost fish Tetraodon nigroviridis reveals the early vertebrate proto-karyotype. Nature. 431, 946-57. 1.1

[6] Megason SG, Fraser SE. (2003). Digitizing life at the level of the cell: high-performance laser-scanning microscopy and image analysis for in toto imaging of development. Mech Dev.120:1407-20. 2