

Master Thesis

Towards Quantitative Spatio-Temporal Gene Expression Measurements in Mice

Thesis Committee:

Prof. dr. ir. M.J.T. Reinders
Dr. ir. B. P. F. Lelieveldt
M. Baiker MSc.
Dr. C. P. Botha
Dr. E. A. Hendriks
Dr. F. M. Vos

Author	Martin Wildeman
Email	m.wildeman@planet.nl
Student number	1047973
Thesis supervisors	Prof. dr. ir. M. J. T. Reinders Dr. ir. B. P. F. Lelieveldt M. Baiker MSc.
Date	January 30, 2009

Abstract

In this project we tried to answer the question whether it is possible to obtain in vivo quantitative spatio-temporal gene expression data of mice, by making use of the GFP protein and Biofluorescence Imaging. The research was driven by a more specific question, being whether it is possible to detect if metastasis to the bone has occurred in cancer progression studies, using GFP cancel cell lines, i.e cancer cells that produce Green Fluorescent Proteins (GFP), at a continuous known rate. We tried to give generic answers by researching the following two subquestions. 1.) Is it possible to register a 3D mouse atlas to 2D Bioluminescence or Fluorescence photographs, based on only those photographs? 2.) Can we make qualitative statements on the location of gene expression, after registration with an atlas?

The first question is covered in a paper called: '*Atlas Driven Registration of 3D Voxel Data to Multi-view Photographs Based on 3D Distance Maps.*' It explains a method to register a 3D piecewise deformable mouse atlas to 2 or more photographic sideviews of a mouse. Based on a distance map that we generated from multiple backprojections of the sideview, we were able to construct an energy function that resembled a 'goodness of fit' of the registration. Comparisons were made with a gold standard and we obtained good results with our method.

The second question is covered in a paper called: '*Testing for Spatial Gene Enrichment in C. Elegans Using Chronograms and a 1D Worm Atlas.*' In this paper we did not focus on the registration process, because this was a straightforward procedure. With an atlas registered to an expression dataset, we applied different statistical tests to answer the question that given the atlas and the expression profile, the observed expression shows enrichment in a selected organ or not. In this paper we show that we are able to filter highly enriched signals out of our complete dataset. We further discuss the added value of our atlas since it is difficult to validate the obtained results of our tests.

Preface

This Master Thesis describes the work that I have done during my graduation project at Delft Technical University in collaboration with the LKEB department of Leiden University Medical Center. It has been a very interesting period in which I have met a lot of new people, got in touch with new areas of science and in which I have learned new things about myself. In good times I enjoyed the gain of experience I gathered. In less good times, I was forced to learn more about myself by doing introspection.

I have a background in both Life Sciences and Informatics and combined this with an interest in image processing. I started this thesis, more than a year ago, with the statement: '*I want to do something in graphics*'. My Professor, Marcel Reinders converted this statement into a new collaboration in which we searched for open opportunities. A result of that for me was an extensive research assignment and the thesis that lies here before you.

I could not have finished this thesis without the help of some people that I want to thank here.

First of all I want to thank my wife, Henrieke, for giving me all the love and support I needed during my graduation project at times I was not motivated anymore. I want to thank my parents, for financing a large part of my study, and for giving me support throughout, especially when I decided to change my study from Informatics to Life Science & Technology.

I want to thank Saxwin Brouwer and Alexander Koppenaar owners of Doxx B.V., for providing me so much flexibility in my working hours during my graduation and for providing me a MacBook that made life so much easier when traveling between Delft and Leiden and enabled me to take my work with me.

I want to thank my room mates in Delft for giving me a pleasant atmosphere to work in. Special thanks go out to Jan Bot, Erik van den Akker, Bart Witteman and Jelle ten Hoeve for sharing thoughts on my work and giving me new insights. Marc Gouw, Michiel Boerhof, Ewine Smits and Peter van Nes provided me with the needed distraction during coffee breaks and during times that I was supposed to work.

I also want to thank my room mates in Leiden, Nora Baka, Vikas Gupta and Meng Ma for introducing a social aspect to my graduation project and the rest of LKEB for providing me with the needed espressos.

Last thanks go out to my supervisors. Marcel for his direct and honest questions and remarks. Boudewijn for always being positive and trying to keep me positive. Martin for giving me help on practical problems and helping me see things in perspective when I needed that.

Martin Wildeman

Contents

Abstract	iii
Preface	v
1 Introduction	1
2 Paper on Registration of 3D Mouse Atlas	3
3 Paper on Statistical Testing of Gene Enrichment in <i>C. Elegans</i>.	23
A Master Thesis Proposal	41
A.1 Master Thesis Proposal	41
A.2 Thesis Research Proposal	41
B Master Thesis Planning Proposal	43
C Working Document on Mouse Paper	45
D Working Document on <i>C. Elegans</i> Paper	85

CHAPTER 1

Introduction

Molecular Imaging can be defined as the *in vivo* characterization and measurement of biological processes at a cellular and molecular level in a noninvasive manner. Molecular Imaging is a relatively new imaging paradigm that instead of looking at macroscopic physical processes, sheds light onto biological processes. It is possible to visualize the transcription rate of genes, by inserting an observable gene, such as Luciferase of the Green Fluorescent Protein which is regulated in the same way as a gene of interest, by placing it downstream of a promoter that is identical to that of the gene of interest¹. With the ability to visualize gene expression the question arises on what can be done with acquired data. To answer this question we take a look into the field of functional genomics, where gene expression data already is analyzed. Most expression data sources do not contain spatial information on expression and in many cases miss a temporal component also. With statistical methods, it is possible to scan for biomarkers and if we have a temporal component in our dataset it may also be possible to reconstruct regulatory pathways or gene networks. With Molecular imaging it is fairly straightforward to acquire a temporal gene expression profile, given that we can compare data obtained in different acquisitions, but Molecular Imaging also introduces a spatial component to the dataset, which is difficult or laborious to obtain with other acquisition techniques currently used in functional genomics. This spatial component might give valuable new information on local gene regulation in an organism and may improve our understanding of cellular processes.

When altering cancer cell lines in such a way that they produce a GFP protein at a continuous rate, these cells can be observed and the number of cells can even be quantified based on fluorescence intensities. In cancer research, Molecular Imaging can aid in progression studies, because it comprises *non invasive* acquisition techniques. Follow up studies of the same organism are therefore possible, which makes it possible to measure and compare progression of the cancer cells over time.

In order to be able to compare expression data from multiple measurements, we must have a method to standardize expression data to some default dimension, size and orientation. One such method is, to be able to map expression data to some predefined (anatomical) 3D atlas. With all expression data mapped to this atlas, we can start comparing signals with each other and start further analysis.

In this thesis project we studied the following two questions. 1.) Is it possible to register a 3D mouse atlas to 2D Bioluminescence or Fluorescence photographs, based on only those photographs? 2.) Can we make qualitative statements on the location of gene expression, after registration with an atlas?

The first question is covered in a paper called: '*Atlas Driven Registration of 3D Voxel Data to Multi-view*

¹Literature Research, Martin Wildeman, 2008

Photographs Based on 3D Distance Maps'. In this paper we have a complex 3D dataset, but we do not have much gene expression data. We therefore focus on the development of an atlas registration in this paper.

We developed a registration algorithm that allows us to register a 3D mouse atlas to 2D photographic projections. This registration procedure will make it possible to map observed expression profiles in the photographic modalities to the 3D atlas. Once a registered atlas is available we can, in principle², calculate the intensity and origin of gene expression and also, more specific, the location and amount of cancer cells inside of the mouse body emitting the photons, when doing cancer progression studies. With localized and quantified expression we can determine whether the observed expression signal is significant, and given the atlas, whether it is 'enriched' at a certain location or not.

Since we did not have expression data for mice available, we used gene expression data of the worm *C. Elegans* to study the second question, in a paper called: '*Testing for Spatial Gene Enrichment in C. Elegans Using Chronograms and a 1D Worm Atlas*'. Since the shape of a worm is simple, the registration step was easy and largely done already, and we could focus on the development and evaluation of statistical test methods.

We compared different statistical tests, that we used to find gene enrichments in our complete dataset. The methods applied on the *C. Elegans* data can, with some modifications, also be applied to expression data from mice, once this data becomes available. We searched for enrichment in qualitative locations, such as 'head' or 'tail'. When using this enrichment detection for cancer progression studies, it may be possible to determine whether cancer has metastasized to a certain organ, by searching for enrichments of Wild e-Man, Afstuderen op 30 januari 10:00, Zaal HB01.010 EEMCS expression in the mouse body. We suspect that the qualitative locations of gene expression may give more insight in the function of genes.

With the evaluation of these two questions we have created a basis on which further analysis on spatio-temporal gene expression data may be applied and in which registration and statistical testing may be merged into one approach.

²A photon propagation model is needed for this. This computationally intensive approach will require a very accurate optical model, as well as a near to perfect registration, otherwise the reconstruction of the photon source will be inaccurate.

CHAPTER 2

Paper on Registration of 3D Mouse Atlas

ATLAS DRIVEN REGISTRATION OF 3D VOXEL DATA TO MULTI-VIEW PHOTOGRAPHS BASED ON 3D DISTANCE MAPS

M. H. Wildeman^{1,2}, M. Baiker², J. H. C. Reiber², M. J. T. Reinders¹, B. P. F. Lelieveldt²

¹Information and Communication Theory Group
Department of Mediamatics
Delft Technical University
The Netherlands

²Division of Image Processing
Department of Radiology
Leiden University Medical Center
The Netherlands

ABSTRACT

In the field of molecular imaging, modalities containing structural data and modalities containing functional data often need to be fused, to obtain (new) biological relevant information. In some cases the modalities that are fused, differ in dimensionality, which makes a 2D/3D registration necessary. In this article we present a new method to register a rigid 3D surface model derived from CT data, or an articulated 3D whole body atlas, derived from the Digimouse dataset, to two or more 2D projection silhouettes, derived from multiplane photography. We show that by making use of a 3D distance map derived from the 2D projection silhouettes, we are able to construct an energy function which can be used for registration. We show that our energy function is able to generate good results for synthetic data as well as for real data with a mean skin surface error of $0.15mm$ and $1.3mm$ distance respectively. We demonstrate the working of this energy function and optimization algorithm by applying it for rigid registration, as well as for a hierarchical registration of the articulated 3D whole body mouse atlas.

Index Terms— Distance Map, Normal Vector, 2D/3D Registration

1. INTRODUCTION

Molecular Imaging comprises imaging of biological processes at a cellular level and at molecular resolution in a non invasive and in vivo manner [1]. A broad spectrum of modalities exists in which each modality can contain different structural and functional information, and when combined, may generate new data which is not available in the separate modalities, e.g. the structural data combined with functional data may provide more information when combined than when studied separately. In some cases, it can be useful to register datasets of different modality *and* dimensionality.

One example of this is the registration of 3D CT or MRI data on 2D bioluminescence imaging (BLI) data, where correct registration can give new insight in the 3D localization and quantification of cancer cells relative to the CT/MRI

dataset, based on the BLI dataset, containing the bioluminescent cancer cell data [2].

Another example is, that because of the non invasive nature of molecular imaging, follow up studies of the same organism are possible, which gives a need for registration of data from multiple experiments, which for example enables the monitoring of tumor growth. It is also of great value to be able to compare biological information, retrieved from intra subject experiments, to be able to measure and compare functional biological data quantitatively.

The goal of this work is to develop a new accurate registration algorithm, to register 3D surfaces to 2D projection silhouettes, derived from multiplane photography. In some cases where registration is needed, only few 2D images are available to obtain a registration. The 3D skin surface can be a rigid body when both datasets contain the same subject having exactly the same conformation or a 3D non rigid deformable atlas. Because we have sparse data with a minimum of only two 2D contours, we need a way to reduce our set of parameters in the non rigid registration process. A hierarchical piecewise optimization seems well suited for this because we will optimize at most 9 parameters at a time. Since we only allow for deformations in the skin surface based on restricted skeleton movement, we only obtain biologically plausible registrations.

The contributions of this work are as follows:

- We introduce a registration energy function that is based on a 3D distance map and includes angle penalties based on the direction of the gradient in this distance map.
- We apply this energy function to register CT datasets or an articulated 3D atlas on multiple 2D contours, using a Similarity transformation or piecewise hierarchical Similarity transformations, respectively.

2. BACKGROUND

Much work related to 2D/3D registration has been reported, describing varying shape representations and optimization criteria. X. Huang [3] gives a nice comparison of different techniques that can be used for shape registration. In all shape registration problems three choices have to be made. First, the form in which the shape is represented has to be chosen. Second, the kind of transformation has to be chosen in order to register the 3D surface on the 2D projections. Lastly, some registration criterion has to be found in order to find an optimum. Huang et al. [3] make use of two distance maps, both for global registration, using mutual information and for generating a free form deformation mesh for registration of non-rigid deformations. The method of Huang seems very well suited for free form contour matching, but we are actually not interested in free form matching, because we want to extend our method to a model with an articulated skeleton and estimated skin surface, which allows only for restricted deformations. Huang, moreover, will need two complete surfaces or contours for their registration, which we do not have and therefore we can not apply their algorithm.

Iwashita et al. [4] generated a 2D distance map based on the estimated contour of a 2D projection from a 3D object. The distance map was used to estimate translation as well as rotation, based on distances and force fields, to register the same 3D rigid body that was used to generate the 2D projection. Scaling was not estimated, because based on only one distance map, it is not possible to determine if scaling in the projection is caused by a large distance in the direction of the camera view, or by real scaling of the 3D object. They generated a 2D distance map, based on a detected contour of a 2D image. They then iteratively calculated new contours of the 3D object and also calculated 'driving forces' for the optimization. These forces are based on the distance and the direction in the distance map gradient. This is a costly approach, since for each iteration the points in 3D space that correspond to the detected 2D contour need to be determined. They managed to effectively determine these correspondences by using dedicated hardware, but if it is possible to avoid switching between a 2D modality and a 3D modality this correspondence problem can be avoided. The approach of Iwashita et al. only works for rigid bodies and, since we want to extend our method to a semi non rigid skin surface model, their approach is not suitable for our goal.

Papademetris et al. [5] make use of an iterative closest point matching algorithm, to register two 3D models. They used a piecewise rotational model, based on an articulated lower body atlas. The skin surface is smoothly updated by making use of a weighted rotation based on the location of the skin vertices, relative to the rotation point. In this way they are able to obtain a smooth, natural skin deformation. For the registration they have a complete lower body 3D skin surface to register to, as well as a complete 3D articulated lower body

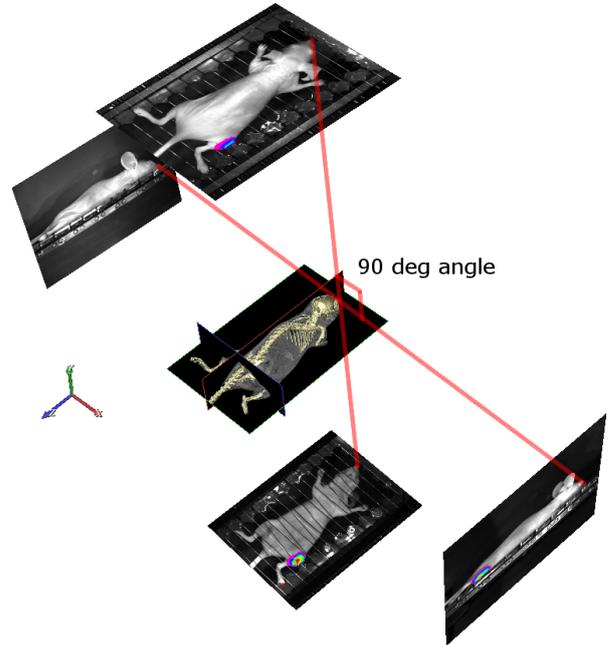


Fig. 1. A visualization of a pre-registration, generated with the Cyttron Visualization Platform, is shown. Four BLI photographs with known reference (angles) between the images, registered to a thresholded 3D CT voxel set are shown.

atlas. We do not have a complete 3D surface to register to and therefore we use an implicit contour representation for the optimization, because recovery of point correspondence would be a computationally expensive task and not straight forward when only making use of two 2D projections [6].

3. METHODOLOGY

Our goal is to register a 3D structural dataset to a set of 2 or more photographic images of the same structure with known reference between the images (See Fig. 1 for an example). In this section we present the methodology we use in our registration algorithm. We start by explaining our shape representation and how we obtain it, followed by the definition of our energy function. We end this section by explaining the optimization algorithm, and we show the different transformations that we apply for registration. Figure 2 gives a schematic overview of the complete algorithm.

3.1. Shape Representation

With our 2D/3D registration we start off with two modalities, one being a CT voxel dataset and the other consisting out of a set of n ; ($n \geq 2$) 2D images of the skin surface, rotated by known in-between angles. A pre-registered example of the two modalities is shown in Figure 1.

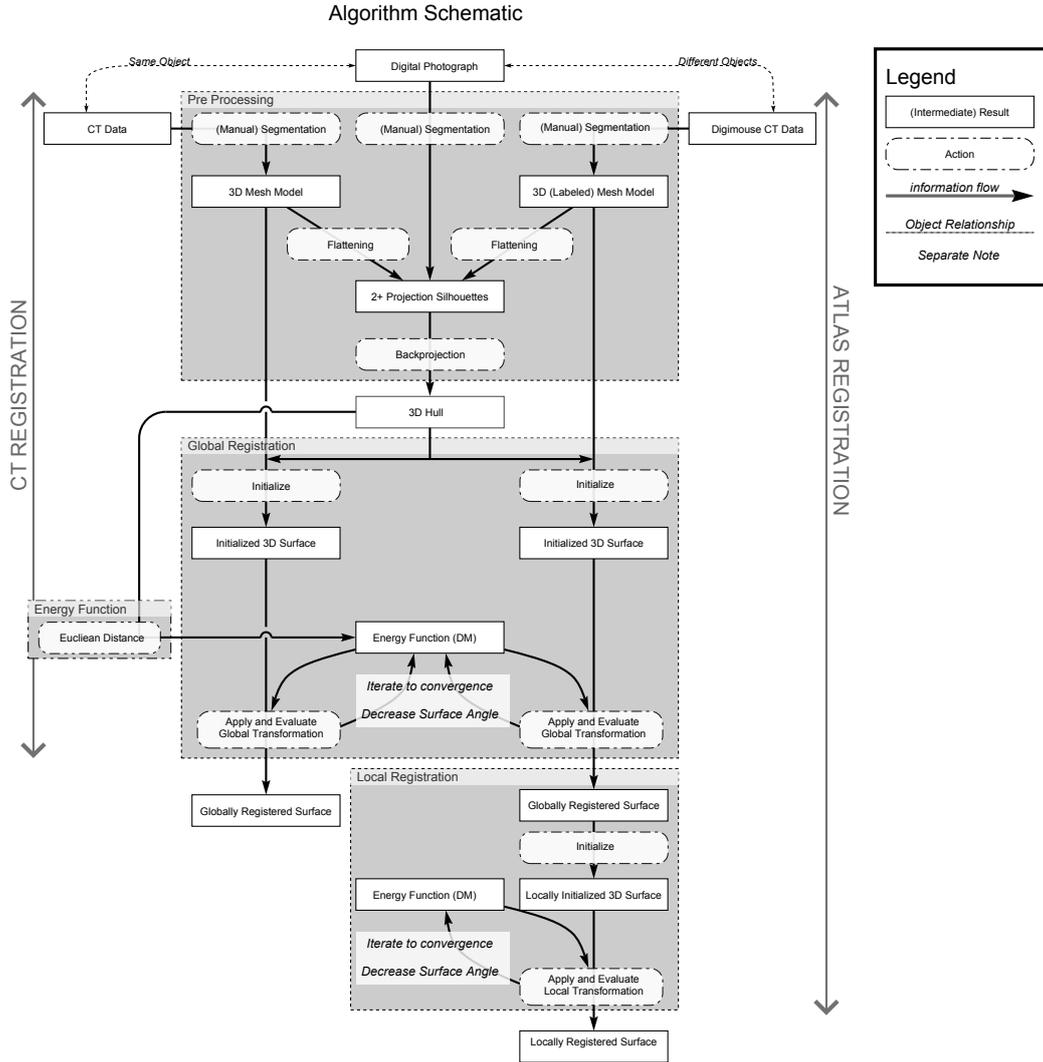


Fig. 2. A schematic of the algorithm we propose. With the obtained 3D surface an energy is calculated. With an optimization algorithm, the 3D surface is transformed iteratively, to minimize the energy function until convergence, to finish with a transformation matrix that defines the obtained registration.

3.1.1. 3D Structural Data

The 3D data can be of any modality, as long as a surface mesh model can be extracted from it, which is needed for the optimization. The source skin surface, i.e. the surface that is extracted from the 3D structural data set, is represented as a triangulated 3D mesh model. Vertices in this model will be referred to as $v \in \mathbb{R}^3$ in this paper. During registration we either use a rigid CT skin surface for the rigid registration or an articulated atlas for non rigid registration. Both the CT skin surfaces as the articulated mouse atlas skin surface are represented as a mesh model. Next to the skin surface, we also segmented the skeleton of the 3D atlas data, resulting in 14 manually labeled bones. These rigid bodies are needed for the hierarchical transformations we propose.

3.1.2. Multi 2D Optical Data

The target surface, i.e. the surface to which the 3D source surface has to be registered, is not fully known. We therefore have chosen to construct an implicit shape representation, based on a limited number of views. To do this, a convex hull volume is created based on the shapes in the 2D images, by back projecting the segmented skin silhouettes. This results in a binary voxel model box, which we define as B , in which all voxels that fall inside the convex hull are assigned 1, and all others 0. The 2D images used are assumed to be orthographic projections which makes back projection a straightforward procedure. A 2D projection of 1D shapes is shown as illustration in Figure 3

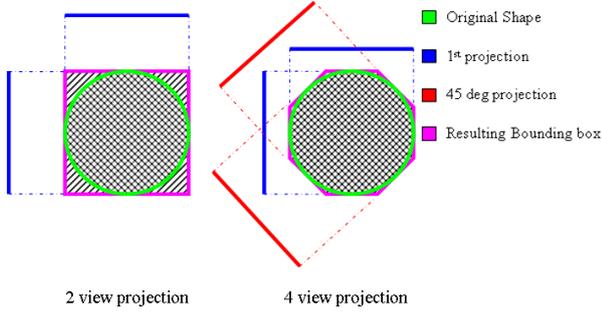


Fig. 3. Schematic of projections as implemented in the algorithm

3.2. Defining the 3D Atlas

3.2.1. Joint Articulation

We created our articulated version of the Digimouse mouse atlas similarly to the method of Baiker et al. [7]. We manually selected pivot points and rotation axes, which we defined by unit vectors \hat{u}, \hat{v} and \hat{w} for the x, y and z axis respectively. The x-axis vectors were chosen to point in direction roughly parallel to the longest axis in the bone. The y-axis was set to point to the joint origin of its parent, or as much as possible in the direction of the y-direction of the world axes, when no parent joint was present. The y-axis of the right ankle joint for example (visualized in Figure 4) points toward the joint of the right knee and is defined as

$$\hat{v}_{\text{init}} = \frac{\text{joint}_{\text{knee}} - \text{joint}_{\text{ankle}}}{\|\text{joint}_{\text{knee}} - \text{joint}_{\text{ankle}}\|}$$

To obtain a 90° angle between the \hat{u} and \hat{v}_{init} vectors, \hat{v}_{init} was corrected by solving

$$90 = \frac{360}{2\pi} \cos^{-1} \left(\hat{v}_{\text{init}} \cdot \frac{\hat{v}_{\text{init}} + n\hat{u}}{\|\hat{v}_{\text{init}} + n\hat{u}\|} \right)$$

n was determined iteratively.

$$\Rightarrow \hat{v}_{\text{corrected}} = \frac{\hat{v}_{\text{init}} + n\hat{u}}{\|\hat{v}_{\text{init}} + n\hat{u}\|}$$

Finally, \hat{w} was obtained with $\hat{w} = \hat{u} \times \hat{v}_{\text{corrected}}$.

3.2.2. Skin Segmentation

To be able to deform the skin surface based on the skeleton movement, we updated the 3D skin surface mesh model by assigning each vertex in the skin surface a label, corresponding to the label of the closest vertex in the skeleton surface. The result of the articulation and skin segmentation is shown in Figure 5.

3.3. Updating the Conformation of the Atlas

To reduce the degrees of freedom (DoF) in our optimization function, we make use of the rigid nature of the mouse skele-

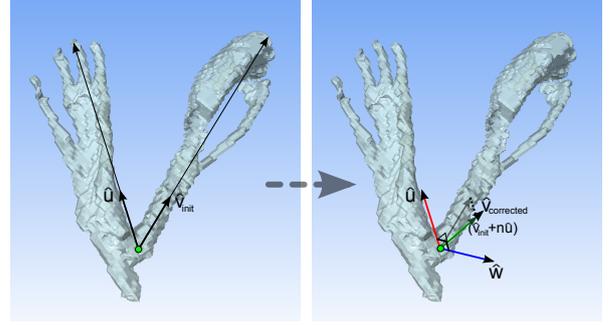


Fig. 4. Visualization of joint articulation. \hat{v}_{init} is corrected, so that the angle between \hat{u} and $\hat{v}_{\text{corrected}}$ is 90° . \hat{w} is defined by the cross product of \hat{u} and $\hat{v}_{\text{corrected}}$. Note that $n < 0$ in this example.

ton, which determines most of the mouse skin shape. Our articulation and joint definitions are similar to what is described in [7], with two main differences. We do not allow translation for each rigid body, but only for the whole skeleton, with its center of gravity (CoG) and rotation axes defined in the 'Spine', as this rigid body is the highest in the hierarchy of the skeleton. All transformations applied to 'Spine' will be applied to bones lower in the hierarchy as well. We make use of uniform scaling for all limbs (1 DoF), instead of the 3 DoF's for scaling defined in [7].

Joint Type	Joints Modeled	DoFs of the articulated bone
Global Orientation	CoG	T_x, T_y, T_z S_x, S_y, S_z R_x, R_y, R_z
 Ball joint	Neck Shoulder Hip Wrist Ankle	R_x, R_y, R_z S_{xyz} (Isotropic)
 Hinge joint	Elbow Knee	R_z S_{xyz} (Isotropic)

Table 1. Joint types used in the articulated atlas. (Images obtained from [8])

To transform all articulated bones and corresponding skin vertices, we make use of one transformation matrix for each bone plus one matrix for global transformation, resulting in a total of 14 matrices, which are obtained by applying a parameter set of $T_x, T_y, T_z, R_x, R_y, R_z, S_x, S_y$ and S_z on the Spine (Global Registration), and R_x, R_y, R_z and S_{xyz} on all other body parts (Local Registration), where the DoF of R_x and R_y are set to zero for the knee and elbow joints (Table 1).

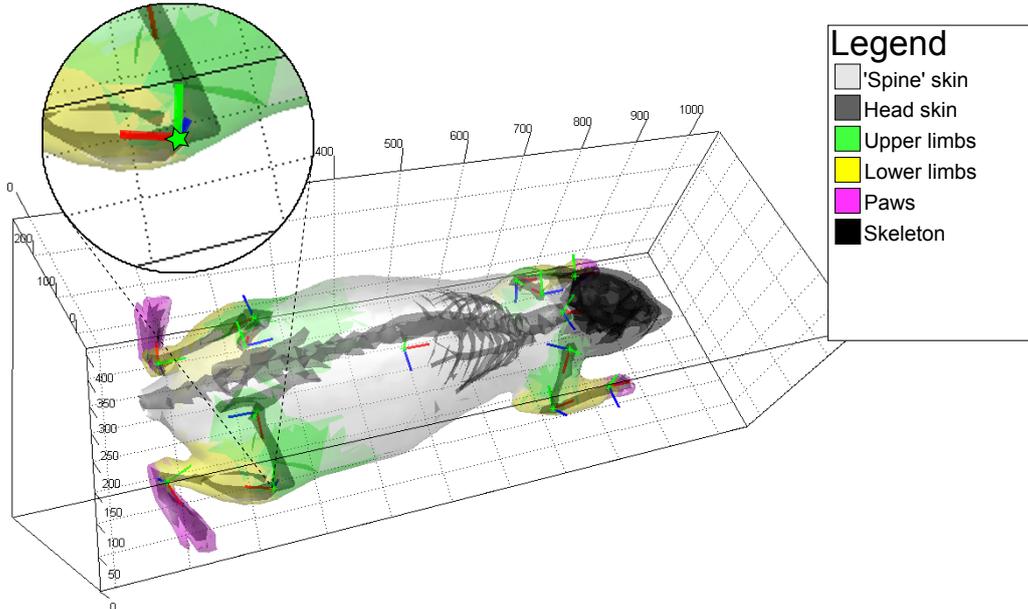


Fig. 5. The atlas we used is based on Digimouse, combined with a CT dataset of mouse feet. The locations and rotation angles of the joints in the atlas were manually selected. \hat{u} is visualized in red, \hat{v} in green and \hat{w} in blue. All unit vectors were multiplied with 40 and have the joint locations as their origin, for visualization purposes. The units of the axis represent the voxel location in the original CT dataset having a voxel size of $100\mu\text{m}$.

The total parameter set Θ consist of 3 translation parameters, 8 scaling parameters (we used 1 scaling parameter per limb) and 34 rotation parameters. A total of 45 parameters can be changed to alter the conformation of the skeleton and the skin, which is transformed with the same mapping function as used for the skeleton, in correspondence with the defined labels of the skin surface vertices. Rotation is achieved using transformation matrices based on unit quaternions [9]. The skin surface is updated with the same transformation matrix as is used for the corresponding articulated bones. The elasticity of the skin is not taken into account and the skin will 'fold' or 'stretch' at the rotation points, as is described in [5]. We expect that the error generated by those vertices, since they are small in number, does not have a large effect on the location of the optimum in the registration process.

3.4. Updating the Orientation of the Rigid 3D CT Surface

The CT dataset is registered in the same way as the global transformation step of the 'Spine' in the mouse atlas. The only difference is how the local axes of the CT skin surface are defined. The mean location of all skin vertices is defined as CoG (the axes origin of the body), and the directions of the local axes are parallel to that of the world axes. Since the CT data and the optical data will be acquired from the same individual, we only apply isotropic scaling, resulting in 7 optimization parameters for the registration process.

3.5. The Energy Function

3.5.1. The Distance Energy Component

To be able to use the binary voxel volume in our registration process, we designed an energy function that allows the minimization of the distance of all vertices of the source skin surface (S) to contour of the voxel set $B = 1$ (As defined in Section 3.1.2). The energy function we propose, is a sum of squared errors (SSE) of all vertices in S and is defined as follows

$$E_{\text{total}} = \sum_{v=1}^n E(v, \Theta)^2 \quad (1)$$

For each vertex v in S an error is calculated as defined in Equation 2, S containing n vertices. Θ is a vector containing all transformation parameters used to register S to B .

$$E(v, \Theta) = DM(\mathbf{x}) \quad (2)$$

$$\mathbf{x} = [T(v, \Theta)] \quad (3)$$

In this equation T can be seen as a mapping function to transform the source shape to the target shape. This can be a single transformation matrix in case of a global affine transformation, constructed with a parameter vector Θ . In case of local transformations this function can be more complex, but is still based on a set of parameters Θ .

3.5.2. Calculating the Distance Map

The distance error function DM is defined by Equation 4 as described in [3]. All remaining occurrences of \mathbf{x} in this paper are defined by Equation 3.

$$DM(\mathbf{x}) = \begin{cases} -D(\mathbf{x}, S_{BE}), & \text{if } \mathbf{x} \in BE \\ D(\mathbf{x}, S_{BE}), & \text{if } \mathbf{x} \in B - (BE + S_{BE}) \\ 0, & \text{if } \mathbf{x} \in S_{BE} \\ \max_{\mathbf{x} \in B} (D(\mathbf{x}, S_{BE})), & \text{if } \mathbf{x} \notin B \end{cases} \quad (4)$$

In this equation BE is defined as the eroded voxel set of voxel set ($B = 1$) using one iteration and a diamond-shaped structuring element. S_{BE} is defined as the set of voxels $((B = 1) - BE)$, i.e. the surface of BE . $D(\mathbf{x}, S_{BE})$ is the Euclidean distance function of a given vertex to the nearest voxel in set S_{BE} . \mathbf{x} is defined as the location $[x, y, z]$ in the 3D voxel volume B . Figure 6 shows a schematic of the voxel set definitions.

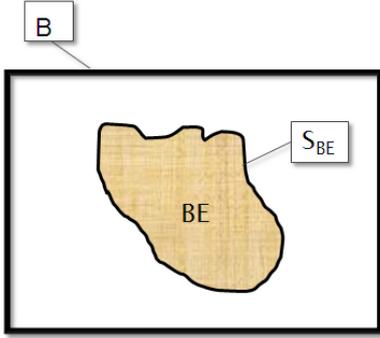


Fig. 6. Schematic of distance map voxel set definitions.

For computational purposes we precalculated the function DM for all possible rounded vertex locations \mathbf{x} where $\mathbf{x} \in B$. This pre-calculation is achieved by calculating the euclidean distance map (DM) of B , using Danielsson's method [10]. The interior (BE) of the contour is assigned negative values, the exterior ($B = 0$) positive values. The voxels belonging to the surface $((B = 1) - BE)$ of the projection are assigned 0. It should be noted that due to the discrete nature of the pre-calculated distance map, the energy function that is obtained is not a continuous function, which has to be taken into account when choosing an appropriate optimization algorithm for optimizing this energy function.

3.5.3. Handling missing features and possible outliers

While the distance map itself gives a good result for a 3D skin surface model that exactly fits in the binary bounding box i.e. when the bounding box has the same shape as the skin surface, it becomes less robust when this is not the case.



Fig. 7. A photograph of a mouse (a) was manually segmented into a binary contour (b). (c) shows a simulated projection of the 3D surface obtained from a CT of the same subject.

The simulated projection generated from a CT skin (Fig. 7c), for example, does not contain skin of the ears, resulting in a significant change of shape compared to the manually segments 2D projections (Fig. 7b). It can also be seen that during acquisition the tail of the mouse has been moved, resulting in variance when registering the two modalities. The SSE is greatly affected by vertices that are far away from the desired skin location and the optimum will have a tendency to shift toward the 'outliers', which could for example be artifact objects obtained when thresholding the CT dataset incorrectly.

Bounding the distance map

To cope with outliers, the energy function needs to be altered, so that outliers are suppressed. This can be achieved by introducing a maximum error D_{\max} , for all vertices generating an distance error of D_{\max} or more. This is defined in Equation 5.

$$DM_{\text{bound}}(\mathbf{x}) = \max(\min(DM(\mathbf{x}), D_{\max}), -D_{\max}) \quad (5)$$

When looking at a visualization of the energy function in Figure 8, it becomes clear that a reasonably good fit is needed first, before this bounded distance map can be used. This is caused by the discontinuity of the SSE function, at the location where the function energy reaches the value of D_{\max}^2 (See the red flat top in the energy function in Figure 8). Equation 1 is affected little by these discontinuities, because it is a weighted average of all vertex energy values ($E(v, \Theta)$) in most cases, but it is affected when it is far away from the optimum. In that case the derivative of E_{total} will be zero for all parameters so that an optimization based on derivatives will fail. Therefore a rough registration can be done first, by using a larger D_{\max} , or by making a good initialization.

3.5.4. Correcting for inherent overestimation of Scale parameters

When minimizing the energy function as shown in Equation 1, the skin surface scaling will be overestimated. When all

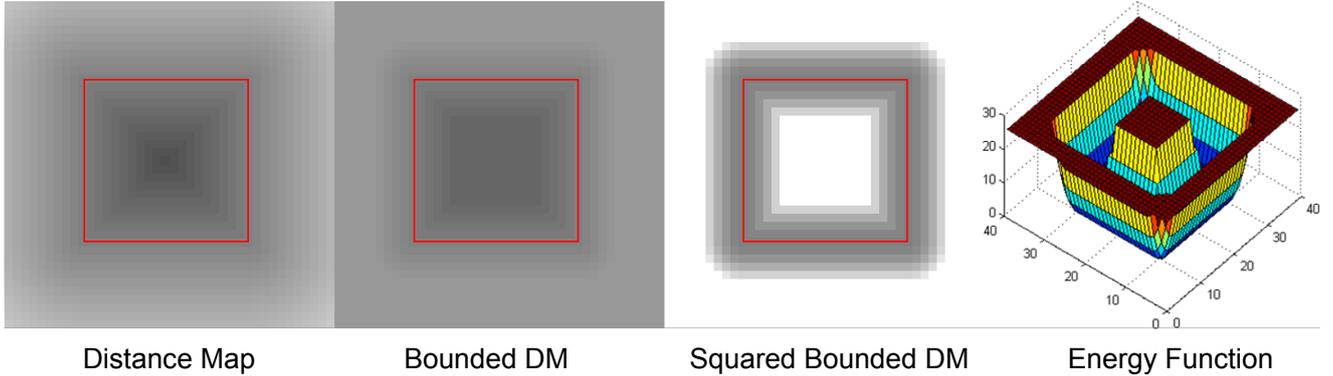


Fig. 8. An example of a distance map and corresponding energy function, based on a 2D Projection of a square (visualized as a red line), which could have been generated by a 2 view orthogonal projection. The distance map was bounded at $D_{\max} = 5$. The squared distance map (DM_{bound}^2) has a maximum value of 25 and a minimum of 0 and is shown as 2D image and a 3D plot of the energy function landscape. The squared DM was pre calculated for efficiency reasons, since $E(v, \Theta)^2 = DM_{\text{bound}}(\mathbf{x})^2$. (Eq. 2)

vertex errors are minimized, the vertices will be located as close to 0 as possible, resulting in a set positioned outside the convex hull, and a set positioned inside the convex hull, so that the SSE is minimized. The best registration would be when all vertices were located inside of the convex hull, as close to 0 as possible, yielding a higher SSE, than the global optimum. The reason for this is that only a few vertices of the 3D body are responsible for the 2D projections observed, but *all* vertices are taken into account when taking the distance map as error measure alone, as is depicted in Figure 9. We need to correct for this error in the energy function, and we introduce two penalties to do so.

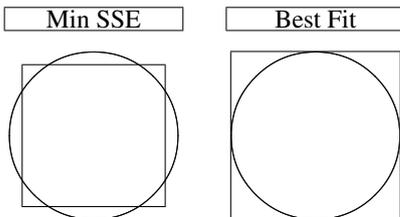


Fig. 9. The squares in this figure represent the bounding boxes. The circle represent the registered source shape. The minimum SSE will not yield the best registration, because it does not fit in the bounding box calculated from the 2D projections which is the goal. All vertices will generate an error and so the optimum will be at the point where all vertices are located as close as possible to the boundary. As a result some vertices will need to be outside the bounding box and some will need to be inside of it.

Penalizing Scaling Overestimation

Since we know that the bounding box is generated by the outer vertices of the 3D skin surface only and therefore vertices that lie outside the bounding box are wrongly positioned, we included an extra penalty α to penalize all vertices located outside of the bounding box, by changing Equation 5 into Equation 6. To have a correcting effect for scaling, $\alpha > 1$

has to hold, and α can not be too large, because real outliers would get a too large effect on the position of the optimum.

$$DM_{\text{boundOP}}(\mathbf{x}) = \begin{cases} \alpha DM_{\text{bound}}(\mathbf{x}), & DM_{\text{bound}}(\mathbf{x}) > 0 \\ DM_{\text{bound}}(\mathbf{x}), & DM_{\text{bound}}(\mathbf{x}) \leq 0 \end{cases} \quad (6)$$

Angle incorporation

Also to correct for overestimation as well as for shape differences between the modalities, we incorporate a conditional penalty in the energy function, which causes the value to be set to D_{\max} , when the angle r_v (Eq. 7) between the skin surface normal of a vertex and the 3D distance map gradient (DMG) at that location is not below a preset maximum angle r_{\max} .

$$r_v = \frac{360}{2\pi} \cos^{-1} (DMG(\mathbf{x}) \cdot \text{vertexnormal}(T(v, \Theta))) \quad (7)$$

$$DMG(\mathbf{x}) = \left\| \begin{bmatrix} \frac{\partial DM(\mathbf{x})}{\partial x} \\ \frac{\partial DM(\mathbf{x})}{\partial y} \\ \frac{\partial DM(\mathbf{x})}{\partial z} \end{bmatrix} \right\| \quad (8)$$

When r_{\max} is set to a smaller value, less vertices will belong to the allowed angle domain. This is visualized in Figure 10. This approach results in an energy function that will decrease when more vertex normals are aligned with the gradient directions in the distance map. This will yield an energy function that is optimal when the skin normal directions are aligned with the gradient directions of the distance map. Note

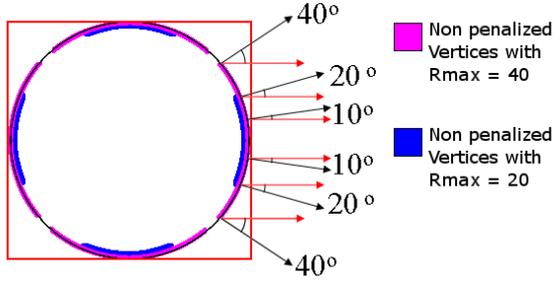


Fig. 10. When r_{\max} is set to a smaller value, more vertices get penalized and thus set to the value D_{\max} . This will lead to an increase in the total energy function. As a result, the energy function will lower, when more skin surface normals are pointing in the right direction, which can be achieved by optimizing the rotation parameters. In the figure above, areas are highlighted which account for the skin surface distance error, i.e. the main part of the energy function that needs to be minimized. In this picture, the red arrows show the direction of the distance map gradient. The black arrows show the direction of the surface normal direction.

that in order for this angle penalty to work, a bounded distance map will be needed during calculation, otherwise the maximum value of the distance map would give too much weight in the final SSE Energy function, resulting in local sub optima (Equation 9).

3.5.5. Driving Force toward Optimum

When the distance map is not bounded, or when D_{\max} is set to a high enough value, the angle penalties cause another effect, which is shown in Figure 11. When the 3D skin surface is on the 'wrong side' of B, this would yield a local optimum at a point where the SSE is minimal. When r_{\max} is set small enough though, vertices on the wrong side will give D_{\max} as error (shown in red), independent of their real distance. This will result in a derivative of 0 for all parameters, until their orientation becomes correct. The green arrows depict vertex normals of vertices that do not get penalized and thus have a value of their real distance. The derivative for all parameters will be based on DMG resulting in a force (blue arrows) on those vertices in the direction of the rigid body. To make sure that scaling is not set to zero, constraints on the parameters are required.

$$E_{\text{angle}}(v, \Theta) = \begin{cases} DM_{\text{boundOP}}(\mathbf{x}), & \text{if } r_v < r_{\max} \\ \alpha D_{\max}, & \text{if } r_v \geq r_{\max} \end{cases} \quad (9)$$

3.5.6. Final Energy Function

Equation 10 shows the final SSE penalized energy function that needs to be optimized for registration.

$$E_{\text{final}} = \sum_{v=1}^n E_{\text{angle}}(v, \Theta)^2 \quad (10)$$

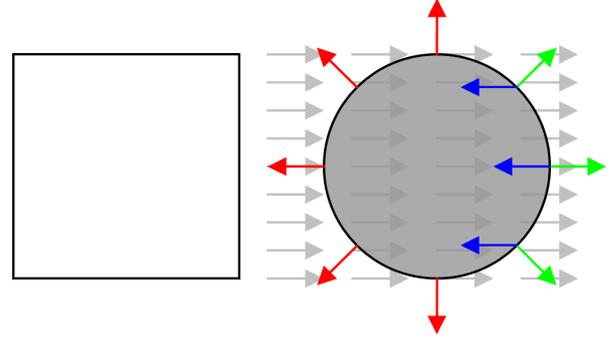


Fig. 11. When making use of angle penalties (r_{\max} is set to 90 in this picture), wrongly initialized surfaces can still be forced to the correct direction. In this image, the gray arrows depict the distance map gradient direction, generated from the square projection. The red arrows depict the vertex normals on the surface causing a penalty. The green arrows depict vertex normals of vertices that do not get penalized. The blue arrows depict the direction in which the non penalized vertices are forced.

3.6. Optimizing the Energy Function

3.6.1. Using the distance map and its gradient

The goodness of fit is determined by calculating an error and is derived from the 3D distance map (DM) and the 3D distance map gradient (DMG), depending on the 3D vertex position. At each iteration of the optimization, Equation 9 is evaluated. In this equation, r_v is calculated for each vertex by applying Equation 7.

3.6.2. Initialization

The first step in the registration process is to provide the optimization function with an initialization. We initialized our model, based on the assumption that the orientation of the input geometry is roughly known. We use the center of gravity (CoG) of the bounding box, combined with an anatomical landmark, the tip of the nose (ToN), to derive the initial transformation matrix. For this matrix we determine scaling parameter (S_{xyz}), by making use of the distance between CoG and ToN in the bounding box and correcting this for the 3D source skin. We also estimate initial translation (T_x, T_y, T_z), by correcting the difference between the CoG of the bounding box and the CoG of the source skin. The last parameter we estimate is the rotation parameter around the y-axis (R_y), by calculating the angle between the xz-components of the vectors $ToN - CoG$ of the bounding box and the source skin. This rotation calculation is based on the fact that mice are positioned on a mouse holder during acquisition and thus variance will be largest in the y axis rotation. The y-axis is defined as the vertical axis in the coronal view.

3.6.3. Global Registration on Photographic Data

When having two datasets of the same subject, acquired with different modalities, registration can be achieved by estimating one single transformation matrix. In this matrix rotation, scaling and translation of the rigid object are implicitly defined.

In Equation 1 every vertex is transformed with parameter set Θ . Θ contains the transformation parameters $T_x, T_y, T_z, R_x, R_y, R_z$ and S_{xyz} , where T is a translation on an axis, R is a rotation around an axis and S is scaling along an axis. The analytical shape of these functions, i.e. which energy of the vertex belongs to which parameter set, is not known, and has to be calculated individually for each vertex. For every possible feasible rounded location obtained with $T(v, \Theta)$ a corresponding distance to B is calculated beforehand and stored in DM . For rigid registration, the rotation parameters are related to the world axis, i.e. R_x means a rotation around the x axis, when the CoG is translated to the origin of the world axes.

3.6.4. Articulated Atlas Registration on Photographic Data

While global transformations alone can be useful to register a 3D volume set to multi plane photography data, we primarily use global registration to demonstrate the correct working of our algorithm and energy function, so we can apply it to the deformable atlas we created.

The notion of a hierarchical transformation model gives us the ability to separate the optimization problem into multiple subproblems, reducing the number of parameters to be optimized in each step and thus reducing the dimensionality of the optimization. To be able to use a hierarchical approach, an energy function is needed that is able to take predefined body part labels into account, while ignoring others. When searching for an optimum of the spine for example, the optimum must not suffer from parts that are lower in the hierarchy because that would affect the optimal fit of the registration. Figure 12 shows the hierarchy we applied in our registration algorithm.

Global Registration Phase

During the registration of the spine and the head, all vertices that belong to the limbs are removed from the energy function, to prevent errors caused by non correctly positioned limbs. The angle r_{max} is decreased during registration, to obtain a better fit and orientation. Once the spine and head are registered and no improvement can be gained, the limbs are registered in a sequential order. Since the conformations of the limbs are independent of each other, the order in which they are registered is not of importance. For the initial registration steps of the spine and the head, DM can be used, instead of DM_{bound} , so that the energy function suffers less from discontinuities. To avoid local minima, angle penalties are not considered. α can still be applied to DM , to avoid

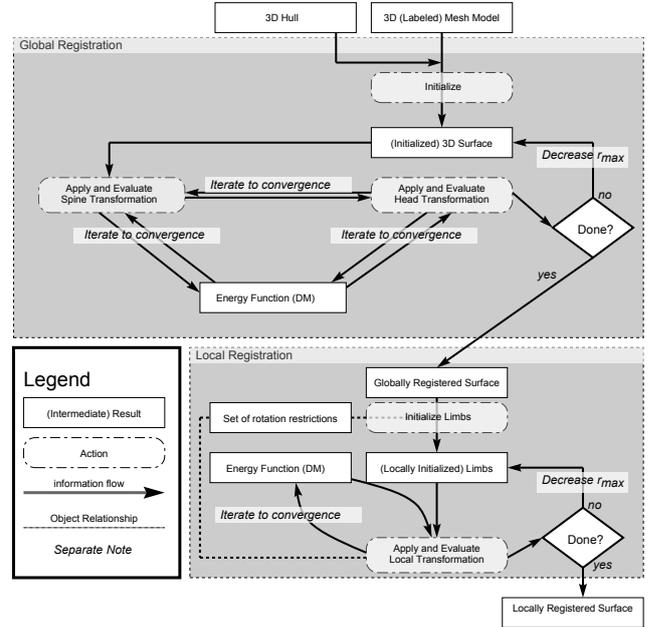


Fig. 12. A hierarchical way of splitting up the transformation as well as the optimization problem.

overestimation of the scale parameter. In this way a rough global registration can be obtained. After this initial registration step, DM_{bound} is introduced and r_{max} is decreased in discrete steps after convergence has been reached.

Local Registration Phase

When the Spine and Head are registered, we start the registration of the limbs by applying a rough initialization of θ_{limb} for each limb, where θ_{limb} contains 7 rotation parameters. For the initialization we do an exhaustive search in the feasible rotation space with an angle increase of s_r and for each feasible orientation ($-R : s_r : R$) we calculate the surface error, including penalties generated by r_{max} . For each evaluation step we set $\theta_{best} = \min(\theta_{best}, \theta_{new})$. We end the initialization when all feasible conformations are evaluated and use θ_{best} as initial parameter set for the registration. The setting of s_r is a trade off between exhaustiveness and calculation speed. A very small step size would indicate a full search space, while a large step size could falsely identify a local minimum as the best initial position possible, with a failing registration as result. A step size of factor $\frac{1}{n} s_r$ would require a factor of n^7 more calculations and thus calculation time. We choose a step size of 10 degrees.

3.6.5. Choice of Optimization Algorithm

For the optimization of the energy function (Eq. 1), we used an iterative non-linear regression method. We chose a Newton interior-reflective method for our optimization, because this algorithm allows us to set upper and lower bound constraints

on the set of parameters out-of-the-box, and it is efficient for medium and large-scale optimization problems [11]. The new locations of the vertices were calculated with each optimization step and the new distances were read from the distance map at each iteration.

4. EXPERIMENTAL SETUP

To test the correct working of our methods we implemented a proof of concept in Matlab 2007b. To validate the algorithm, we designed tests to demonstrate the working of the distance map method by using synthetic data and real data with a known gold standard registration. In addition we designed an articulated mouse atlas from Digimouse which allowed for non-rigid deformations.

4.1. Data Acquisition

4.1.1. Synthetic data

We define synthetic data as real CT data from which the skin surface is segmented, and of which we made projections on a 2D space, to generate synthetic binary 2D skin segmentations. The CT skin surface was scaled and translated in such a way that when voxelizing the volume it would roughly span 300x150x150 voxels. This was done for memory reasons. The surface was then refined by setting the maximum distance between all connected points on the surface to be 1. The 2D projections were generated by rounding the locations of the vertices and setting the corresponding locations in voxel space to 1. A projection silhouette was made in both the x and y direction of the mouse surface, resulting in two binary images of which one is shown in Figure 7c. The two binary contours were backprojected into 3D space, to obtain a 3D binary voxel model, that acts as bounding box for the registration problem. In addition to the existing distance map generated out of two 2D skin surfaces, we also generated a projection matrix based on four 2D skin surfaces, to determine the sensitivity for the algorithm to additional input data and to compare the obtained improvement with the 2 surface case. For the generation of the synthetic side views, i.e. the projection of the 3D skin surfaces, we let the origin of rotation be at the center of gravity of the binary voxel volume as generated from the first two side projections. We then rotated both the voxel volume and the skin surface model around this points' z-axis for 45 degrees, and calculated two new projections resulting in a second voxel volume. We then rotated 45 degrees back, to let the 4 view projection have the same orientation as the 2 view projection.

4.1.2. Real data

The real data that we used is based on the same CT dataset, but now accompanied with real 2D BLI images, in which we *manually* drew contours, to obtain binary 2D skin projections

(Fig. 7b). We only used a top and side view BLI image to obtain two contours. Again for memory reasons, we down-sampled the images to 300x150 pixels. The 3D binary voxel model was obtained with the same method that was used for the simulated projections.

4.2. Test setup

4.2.1. Finding α

When only rigid transformations are applied during initialization and registration, we know that the 3D source skin surface will fit well to the simulated projection space. Because of the fact that we discard information by making projections of the skin surface, the optimum is not located exactly at the original skin surface anymore. For a completely convex model, a perfect fit can in theory be obtained by letting $r_{\max} \rightarrow 0$. Though as r_{\max} converges to zero, at some point not enough vertices are left to generate a useful energy function as virtually all vertices are given the error αD_{\max} . As a result there will always be an overestimation when $\alpha = 1$. Obtaining the right value for α is not straightforward and is dependent on whether 2 or 4 views are used, which sampling is used and also on the absence or presence of concavities in the 2D projections. We know for sure that α must be larger than 1, otherwise the overestimation effect will become larger. To find a good value for alpha, we ran all registrations on all mouse examples, for a range of $alpha = [0.5 \dots 9]$

4.2.2. Rigid Transformation

We used CT datasets with accompanying BLI datasets of 10 different mice. The CT datasets were individually thresholded and converted to a triangulated mesh, to obtain skin surfaces. We also segmented a top view and side view of each mouse, resulting in a total of 20 2D projection silhouettes. The transformation matrix of a pre-registration of the CT and BLI datasets was used as a gold standard to compare our registration with. This pre-registration was acquired by making use of expert landmark annotation in the two modalities and registration of those landmarks. This registration was used as a gold standard. To compare the added value of more than two projections we ran tests with two synthetic projections and with four synthetic projections. We also tested the added value of the incorporation of the angle penalties by running a test without the use of these angles. Finally the effect of α was measured. An overview of the tests for rigid registration is shown in Table 2.

4.2.3. Hierarchical Model

To test the robustness of our energy function and optimization algorithm we generated random deformations in our mouse model, where we changed the size of each bone allowing it to be between 80% and 120% of the original size. We also rotated

α value	r_{\max} values	2 Side Views	4 Side Views	Abbreviation used in Table 3
Synthetic Projection Data				
1	180,45,10,5	Y	Y	Synthetic Data 2/4 surf.
1	180	Y	Y	Synth. 2/4 view w/o angle
2.5	180,45,10,5	Y	Y	Synth. 2/4 view outer pen. 2.5
Real Data (BLI Segmentation)				
1	180,45,10,5	Y	N/A	Real Data 2 view
2.5	180,45,10,5	Y	N/A	Real Data 2 view outer pen. 2.5

Table 2. Overview of used tests.

each joint within the allowed feasible restriction set. We generated a random deformation with $\pm 10^\circ$ change in all joint rotations, while maintaining the degrees of freedom in rotation. We then generated changes in neck, left arm, right arm, both arms, left leg, right leg, both legs and all limbs to get an indication of robustness of the algorithm. We used 2 and 4 synthetic projections to generate the bounding boxes, as we did for the affine transformation tests, to determine whether extra information on the bounding box would yield better registration.

Since we did not have a gold standard for the hierarchical model registration, we only used synthetically generated projections to test with.

4.2.4. Validation Indices

To measure the quality of the registration we calculated 1) the differences in scaling and translation compared to the gold standard, 2) the distance between two CoG points, 3) the mean surface distance, 4) a mean Dice coefficient [12] and 5) the in between angles of the gold standard registration and our own registration, by applying the transformation matrices to the vector $[1, 1, 1]$ and calculating the dot product between obtained vertices, after correcting for translation.

4.2.5. Used Software

For the manual segmentation of the CT skin surface and the Digimouse 3D atlas we used the software package Amira. For implementation of the registration algorithm we used Matlab.

5. RESULTS

5.1. Results of Rigid registration

In Table 3 we compare results obtained from the synthetic 2 view case, the 4 view case with and without angle constraint and finally the real data 2 view case in comparison with the gold standard, where real data is defined as manually segmented 2D BLI images, to which a 3D skin surface generated from a CT scan is registered. We also show the improvement obtained with incorporating an α of 2.5.

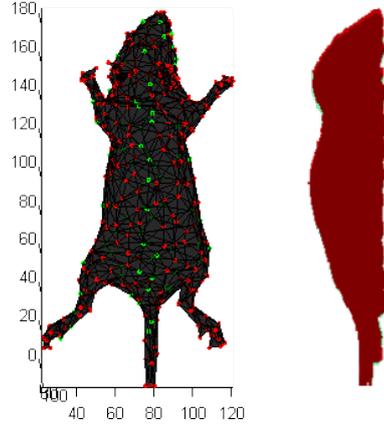


Fig. 13. Vertices on the skin surface that are penalized (red) give an error of 1 in the error function. The other vertices (green) will get their corresponding distance in the distance map. It can be seen that the most vertices that are below the allowed maximum angle (green) are located at the back. These vertices are responsible for the shape of the 2D contour

5.1.1. Results for α

With our 10 real data samples and synthetic projections we found that for two view projections, and α of 2.5 was optimal by comparing performance with varying α , and for four views an α of 5 showed the best average results. The graphs we used to make this decision are shown in Figure 14. As can be seen the values of α do not show a very clear optimum. We have too little data to determine a clear optimum, but we can see that the optimum is not really sensitive to different values. It is clear from Figure 14 that α should be larger than 1.

5.1.2. Synthetic Data

In Table 3 the results of our rigid registration method are shown. With a mean surface error of less than a voxel and Dice indices approaching to 1, we obtain near perfect registrations. This shows that the energy function performs well for rigid transformations.

As can be seen in the synthetic data tests, in the 2 view cases the skin error is slightly smaller compared to the 4 view cases, which is unexpected, since 4 views give more information about the shape of the mouse than 2 views. On the other hand, the variance of the 4 view case is smaller, indicating that the algorithm performs more stable when using 4 views, which can be explained by the extra information we have. Also, the rotation errors indicate that the 4 view cases generate better orientations of the mouse.

5.1.3. Real Data and Gold Standard

In order to validate the working and accuracy of our method, we compared our transformation matrix to one with known

	Distance between Skin Surfaces ¹	Corrected Distance between Surfaces	Distance between Centers of Gravity	Scaling Error	
Unit	voxels	voxels	voxels	%	
Synthetic Data 2 surf.	1.91 ± 1.83	0.92 ± 0.367	1.27 ± 1.22	0.74 ± 0.539	
Synthetic Data 4 surf.	2.17 ± 1.70	1.08 ± 0.146	1.08 ± 0.696	1.33 ± 0.338	
Synth. 2 view w/o angle	3.47 ± 1.38	2.46 ± 0.389	3.21 ± 2.52	2.95 ± 1.2	
Synth. 4 view w/o angle	2.57 ± 1.44	1.61 ± 0.417	1.71 ± 1.1	2.30 ± 0.974	
Synth. 2 view outer pen. 2.5	1.23 ± 1.56	0.353 ± 0.151	0.279 ± 0.235	-0.319 ± 0.220	
Synth. 4 view outer pen. 2.5	1.35 ± 1.67	0.424 ± 0.0762	1.12 ± 0.32	-0.0665 ± 0.186	
Real Data 2 surf.	4.20 ± 1.62	3.96 ± 1.19	11.2 ± 7.47	-1.65 ± 1.53	
Real Data 2 surf. out. pen. 2.5	4.15 ± 1.75	3.86 ± 1.32	9.03 ± 6.20	-3.40 ± 1.86	

	Rotation Error	Dice Coefficient	Translation Error		
Unit	degrees		x voxels	y voxels	z voxels
Synthetic Data 2 surf.	0.592 ± 0.423	0.995 ± 1.51 · 10 ⁻³	1.07 ± 1.68	0.503 ± 1.54	3.76 ± 3.1
Synthetic Data 4 surf.	0.367 ± 0.268	0.995 ± 6.46 · 10 ⁻⁴	3.11 ± 1.44	2.75 ± 1.12	4.77 ± 2.86
Synth. 2 view w/o angle	1.26 ± 0.488	0.988 ± 1.92 · 10 ⁻³	6.21 ± 3.57	7.34 ± 5.25	8.98 ± 5.73
Synth. 4 view w/o angle	0.475 ± 0.188	0.992 ± 1.92 · 10 ⁻³	4.79 ± 3.00	5.20 ± 4.08	8.03 ± 2.95
Synth. 2 view outer pen. 2.5	0.598 ± 0.384	0.998 ± 5.10 · 10 ⁻⁴	-1.25 ± 0.855	-1.17 ± 0.704	1.63 ± 2.08
Synth. 4 view outer pen. 2.5	0.336 ± 0.224	0.998 ± 3.68 · 10 ⁻⁴	0.433 ± 0.800	-0.336 ± 0.695	1.42 ± 1.10
Real Data 2 surf.	1.73 ± 1.22	0.983 ± 3.00 · 10 ⁻³	7.25 ± 6.51	14.5 ± 7.69	1.31 ± 10.5
Real Data 2 surf. out. pen. 2.5	1.66 ± 1.03	0.984 ± 3.07 · 10 ⁻³	-11.7 ± 7.57	17.8 ± 5.11	2.78 ± 11.3

Table 3. Results of different optimization problems, compared to a gold standard generated by an in house developed program. A voxel roughly corresponds to 350 μ m. The mean skin surface error does not provide a good approximation of the real skin distance error. The voxelization of the CT surfaces is restricted to the domain of the BLI data, which is not equal to the CT domain. When calculating the distance of the registered surface to the voxelized surface, the domain of the CT skin surface that lies outside of the voxel domain generates an error. We therefore estimated the real error (second column) by leaving out all vertices located outside of the voxel space.

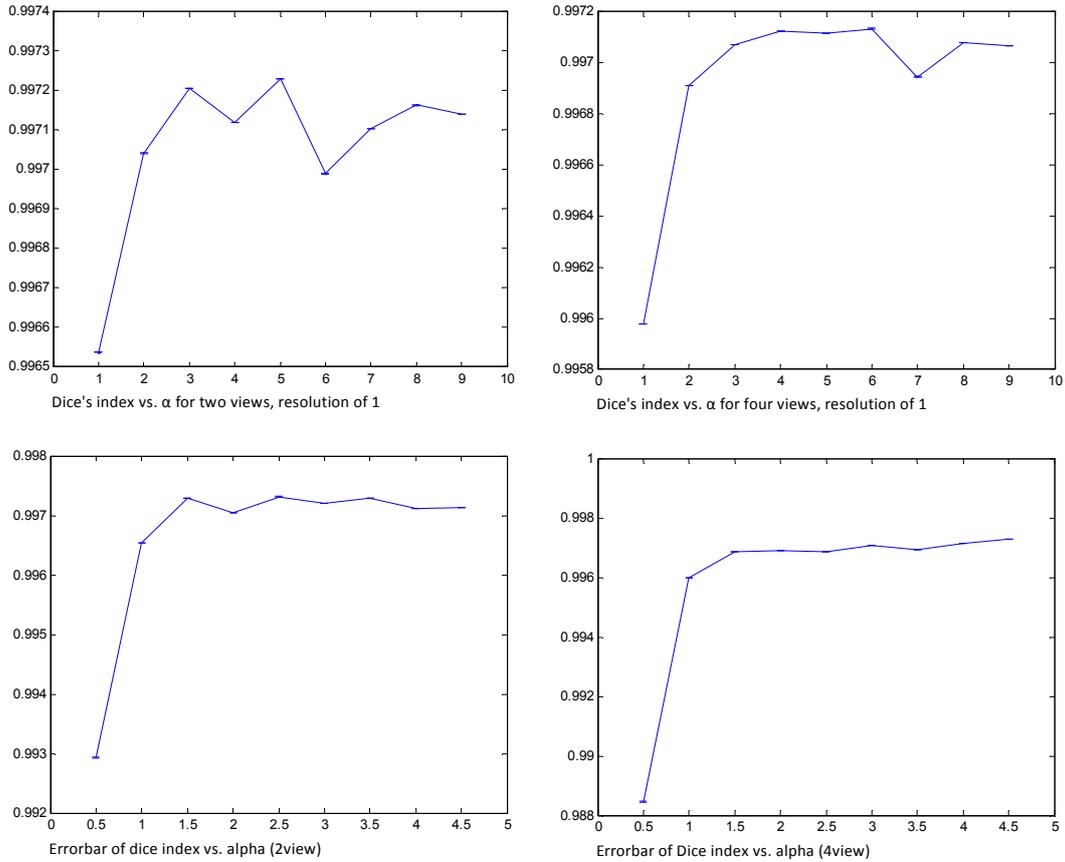


Fig. 14. 4 tests for α with different ranges for α and different number of used side views.

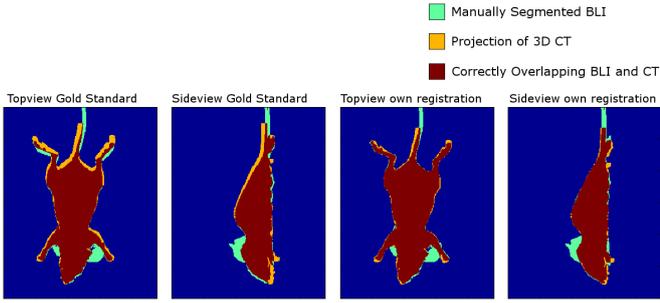


Fig. 15. A qualitative comparison between our gold standard and our own registration.

registration generated by the Cyttron Visualization Platform [13], an in house developed program. The registration in this program was achieved by using expert landmark annotation and point-based registration. Three points were drawn into the 2D images and in the 3D voxel set, and landmark based registration was applied, using Procrustes method.

When looking at the real data performance of the algorithm, we see slightly worse Dice indices than for the synthetic data, but we still obtain an accurate registration with a mean surface error of $\approx 1mm$ and a Dice index of 0.98. The observed increase in error is mainly caused by the differences in contour shapes between the real 2D mouse surfaces and the 2D CT projections. These differences can be caused by loss of information during preprocessing, such as noise reduction or by wrongly segmented photographs due to low contrast or artifacts cause by reflection of light. Also small movements of the mouse between the two acquisition phases are seen.

Based on an observed Dice index of 0.977, we must conclude that the gold standard that we use is less accurate than our automated registration, which is good. As a result though, the errors that are compared to the gold standard (everything except the Dice index) are not as informative as we would like. The errors that we observe for the real data are slightly larger than for synthetic data, but errors are not solely caused by registration errors. A part of the errors is caused by shape differences, as explained in subsection 3.5.3, a part is caused by the registration errors in the gold standard, and a part is caused by the actual registration errors of our own algorithm. An example of a visual comparison between the gold standard and our registration is shown in Figure 15. As can be seen in Figures 7 and 15, a part of the registration 'error' in Table 3, is caused by a difference in shape between the 2D BLI data and the 3D CT data projections. Because of these shape differences, an inconsistency between DM and the CT skin surface is created, which changes the location of the optimum. Moreover the regions between the upper limbs and the ears suffer from non linear gradient fields, which could cause the angle penalty to generate extra errors. Also, the pose of the tail is not completely fixed and thereby generates extra errors. Moreover, because the tail of the CT modality exceeds

the covered space of the BLI modality and also the synthetically generated contour modalities, the part of the tail that is located outside of this modality generates an error. Since the position of the tail of the mouse is probably non relevant information in most cases, it could be taken out of the registration energy function, for example by removing it from the CT dataset. Our visual check explains a large part of the errors we observe and we conclude that the used energy function performs very well, given that the segmentation of the optical data is of high quality.

5.2. Results of Atlas Registration

Because of the many parameters and the hierarchical structure of the Digimouse atlas, it is difficult to interpret all obtained results, especially since errors high in the hierarchy propagate to lower parts in the hierarchy, but are partly corrected (e.g. a wrongly positioned rotation point, will be compensated by a 'wrong' rotation). The Dice index and skin surface errors are a good indication of performance and easier to interpret.

5.2.1. Atlas registration to perturbed atlas projections

Since the algorithm makes use of only side views with a minimum of 2, it is very likely error prone in the case where a mouse has its limbs positioned close to its body. Determinations of rotations in the z-axis would be inaccurate, because of the cylindrical shape of the mouse, and therefore the rotations around the x and y axis could also be erroneous. Since rotations in the x, y and z direction are not independent variables it is of no use to take these parameters as a validation index. Instead we used, as mentioned in Experimental Setup, the inner rotation angle between initialized and registered transformation matrices. Plotting these angles all in the same figure would give an unclear figure, so we merged all rotation errors in a limb and all translation errors in a limb, to a mean rotation and mean pivot distance respectively (Fig. 16E, F). Although we lose information with this approach, it still gives an indication on performance of the registration.

As can be seen, with perturbations in individual limbs, the mean error of the corresponding limb parameters also increases. Due to the hierarchical approach, a rotation error in the Spine will result in a pivot point error and rotation error in the upper limbs. The upper limb rotation errors generate a translation error and rotation error in the part lower in the hierarchy. It is therefore very important to have a good registration high in the hierarchy, because registration of children will fail otherwise. We could also allow small translations for individual articulated bones, to correct for inaccurate registrations, but this would increase the DoFs.

In Figure 16E and F it can also be seen that the left leg registration performs systematically worse than the right leg registration. While searching for an explanation for this error, we discovered that a small error was introduced in the segmentation and labeling phase of the Digimouse CT dataset,

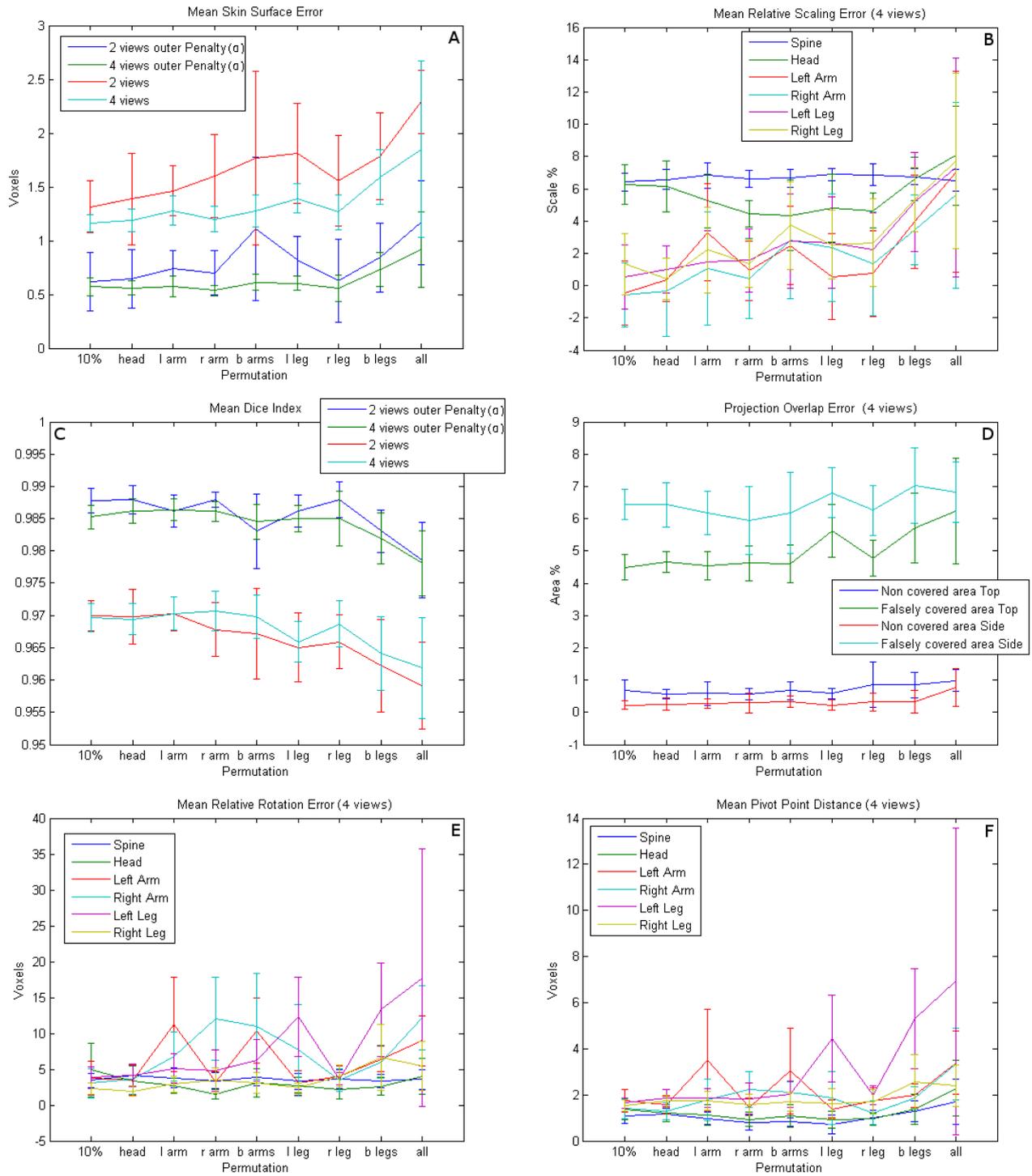


Fig. 16. **A:** Surface skin error after registration for a synthetic 2 view case and 4 view case. With ($\alpha = 2.5$) and without ($\alpha = 1$) outer penalty. **B:** The spine and head were overestimated in size in all cases. The incorporation of an extra penalty α solved this overestimation problem. Registration was done with $\alpha = 1$ and four view projection. **C:** Dice Index after registration for a synthetic 2 view case and 4 view case with ($\alpha = 2.5$) and without ($\alpha = 1$) outer penalty. **D:** The percentage of falsely covered areas are the percentages of the calculated projections of the registered skin surface, that do not overlap with the 2D contour which are used for registration. The non covered areas are the areas of these 2D images that are not covered by the projection of the registered surface. Registration was done with $\alpha = 1$ and four view projection. **E:** rotation errors for spine and limb and average error of all joints per limb. **F:** pivot point distance error for spine and limb and average error of all joints per limb. All registration were run with $s_r = 20$.

which may be causing this error. The left foot and left lower leg are overlapping in this labeling, resulting in a skin vertex labeling that is also wrong, because we make use of a closest point calculation. When applying rotations to the ankle joint, errors will be introduced in the vertex normal directions. We expect that the error penalties resulting from this are causing the systematic error in the registration of the left foot.

It can be seen in Figure 16B and 16D that the Dice indexes are mainly affected by an overestimation in size of the Spine and Head labeled vertices, in spite of the angle penalties used. To compensate for that we used α in the same way as we did during the rigid registration. An improvement can be seen in the Dice indices when we use this penalty, (Fig. 16C) and we observed that the overestimation and false overlap decreased.

With α set to a value above 1 and the use of angle penalties, we obtain a Dice coefficient of 0.98 for all perturbations. Though it should be noted that the conformation of the mouse limbs has a small effect on the Dice index, this result is good. We observe that the pivot point distances of all limbs, except that of the left leg, remain within 3 voxels, which corresponds to $\approx 1mm$.

5.2.2. Atlas registration to CT Projections

We also made some first steps in looking at performance of our atlas registration to CT projections and obtained skin surface errors of 7.42 ± 1.14 voxels and a Dice Index of 0.973 ± 0.003 . These results are worse than what we observed with CT to real data registration which is shown in Table 3, but these registration indices are still good. A visual inspection shows that many cases errors are caused by local sub optima, but we also see that the estimation of the skin surface that we calculate is based on a too simple model. Apart from this, the use of an articulated atlas looks promising.

6. DISCUSSION

6.1. Contour Detection

Our registration algorithm heavily relies on the correct contour detection of the mouse skin surface boundary of 2D skin surface images, for instance BLI images. We have manually segmented our BLI dataset, which was of good quality and allowed us to make a good, although not quantifiable, segmentation. Some contour detection algorithms can be used for this, for example an active contour model [14]. Also the model is known to consist of one part, so a closing and selection can be applied to achieve one binary blob. To make sure correct segmentation is achieved, a mouse holder should be used that does not occlude the mouse. If this is not possible, an iterative method could be thought of that uses the optimization algorithm to find more accurate contours of the skin, to feed the next optimization. For a reasonable segmentation though, we would need photographic data with a high signal (mouse) to noise (rest) ratio. Observed occlusions in

the mouse image data suggests that a shape model will probably be necessary to aid in automatic segmentation. As can be seen in Figure 7a, at the boundaries where the projection contour would be, the mouse skin surface is darker, because of shading. When applied automated segmentation by thresholding, chances are that shaded part of the mouse will be recognized as background, resulting in a smaller segment. This will affect accuracy of the registration, mainly in scaling.

6.2. Vertex distribution on skin surface

The mean skin surface error that is shown in Table 3 is based on a uniform distribution on the skin surface. We used the skin and its projection for registration of a 3D volume. The point distribution versus 3D volume ratio is not constant, but varies along the direction of the spine. It could be argued that, although this uniform point distribution is working well for the registration process, the obtained skin surface error is probably not a good representation of the registration error. Although we are registering a skin surface, our goal is the registration of a 3D volume. A volume overlap error, in the form of for example a Dice index, would therefore be a better measurement.

6.3. Concavities

We make use of a backprojection of projected contour silhouette, to obtain a convex hull as an implicit representation of our skin surface. Since mice are mainly 'cylindrical' in shape, this approach works quite well in our case, but when more concave surfaces need to be registered, our proposed energy function will most likely fail. The angle penalty, set by r_{max} , will increase errors, although partially undone by the bounding of DM with D_{max} . The only setting that can be tuned is α , to compensate the large overestimations of scaling parameters that will be caused by concave surfaces.

7. CONCLUSIONS AND FUTURE WORK

7.1. Conclusions

We presented a novel method to register a 3D skin surface to two or more 2D surface images. We demonstrated the working of this algorithm on synthetic and real data and showed that it is important to have a correct segmentation of the 2D images for the algorithm to work.

We also showed that 2D projections themselves provide sufficient information to obtain a good registration when only rigid transformations are applied, and, surprisingly, that a dataset of 2 projections even outperforms a dataset of 4 projections. The atlas registration is less robust, but we still manage to obtain a mean surface error of 1 voxel when we fully perturb the atlas parameter set, based on 2 projections. The registration of the atlas on real data does not yield the desired results yet, and we probably need a more natural and

flexible skin surface model, that is able to cope with intra subject shape differences, as well as to give a better estimation on shape differences caused by elasticity of the skin surface.

7.2. Future Work

In order to improve the performance of the left foot registration, we need to obtain a better segmentation of the Digimouse atlas. With this new segmentation the experiments of the atlas registration have to be redone to determine whether the segmentation error is indeed causing the observed errors.

While we are currently able to register an articulated atlas on 2D projections, we do not have access to real data with a known ground truth for joint rotations at the moment, which makes it impossible to validate our results. Also we will need a robust method that estimates the location of internal organs correctly, based on the estimated location of the skin and skeleton of Digimouse, to be able to use this technique for quantitative light reconstruction.

We also will need to improve the estimation model of the location and shape of the mouse skin surface, based on the position of the skeleton. We know that the shape of the skin surface at the abdominal region is not well defined by the conformation of the skeleton so a more advanced model is needed to model differences in that region. Although our method currently allows limbs to change conformation in all directions, it is more likely that it will have more restrictions than we currently modeled, since a mouse holder will decrease the feasible solution space and thus the DoFs in limb rotations. These restrictions need to be studied and incorporated into the model, most likely resulting in conditional rotation constraints (and the introduction of new local minima). Note that a mouse holder will probably also have an effect on the skin surface near the abdominal region, because of limiting space in that area.

By using genetic techniques, it is currently possible to insert light emitting or light fluorescent proteins in biological cells. These techniques can be used for keeping track of migration and growth of for instance cancer cell lines, but they may also be used to monitor gene activity inside of an organism, in a spatial-temporal manner [15]. In molecular imaging, to determine the spatial and temporal location of these cells inside a mouse, BLI scans can be made. Since for these scans CCD cameras are used, only pictures of the surface of a mouse can be made. For superficial localizations of the light source, a thresholding on lumen can be made, but for deeper lying regions, a photon propagation model is likely to give a better prediction of the source of light [2]. In order to be able to reconstruct the source of light, the inverse problem has to be solved and for this a 3D model with known optical properties is needed. By registering the surface pictures to, for example, a CT dataset, a reconstruction of the light source can be made. This technique is referred to as bioluminescence tomography (BLT).

If we are able to obtain a robust and accurate registration algorithm, we will be able to map acquired BLI data to the atlas and start making quantitative analyses of observed spatio-temporal gene expression data. For these analyses, statistical tests will have to be developed to make reliable statements on expression levels [15].

8. REFERENCES

- [1] R. Weissleder and U. Mahmood. Molecular imaging. *Radiology*, 219(2):316–333, 2001.
- [2] Ge Wang, Wenxiang Cong, Kumar Durairaj, Xin Qian, Haiou Shen, Patrick Sinn, Eric Hoffman, Geoffrey McLennan, and Michael Henry. In vivo mouse studies with bioluminescence tomography. *Opt. Express*, 14(17):7801–7809, 2006.
- [3] Xiaolei Huang, Student Member, Nikos Paragios, Dimitris N. Metaxas, Senior Member, and Senior Member. Shape registration in implicit spaces using information theory and free form deformations. *IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, 28:1303–1318, 2006.
- [4] Yumi Iwashita, Ryo Kurazume, Tsutomu Hasegawa, and Kenji Hara. Fast alignment of 3d geometrical models and 2d color images using 2d distance maps. In *3DIM '05: Proceedings of the Fifth International Conference on 3-D Digital Imaging and Modeling*, pages 164–171, Washington, DC, USA, 2005. IEEE Computer Society.
- [5] X. Papademetris, D. P. Dione, L. W. Dobruki, L. H. Staib, and A. J. Sinusas. Articulated rigid registration for serial lower-limb mouse imaging. In J. Duncan and G. Gerig, editors, *miccai*, pages 919–926. Part II, LNCS 3750, Springer, 2005.
- [6] Xiaoguang Wang, Yong-Qing Cheng, Robert T. Collins, and Allen R. Hanson. Determining correspondences and rigid motion of 3-d point sets with missing data. In *CVPR '96: Proceedings of the 1996 Conference on Computer Vision and Pattern Recognition (CVPR '96)*, page 252, Washington, DC, USA, 1996. IEEE Computer Society.
- [7] Martin Baiker, Julien Milles, Albert M. Vossepoel, Ivo Que, Eric L. Kaijzel, Clemens W. G. M. Lowik, Johan H. C. Reiber, Jouke Dijkstra, and Boudewijn P. F. Lelieveldt. Fully automated whole-body registration in mice using an articulated skeleton atlas. In *ISBI*, pages 728–731. IEEE, 2007.
- [8] Preventdisease.com. http://preventdisease.com/fitness/fundament/articles/types_of_joints.html.

- [9] Leandra Vicci. Quaternions and rotations in 3-space: The algebra and its geometric interpretation. Technical Report TR01-014, 27 2001.
- [10] P.E. Danielsson. Euclidean distance mapping. *Computer Graphics and Image Processing*, 14:227–248, 1980.
- [11] Thomas F. Coleman and Yuying Li. An interior trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*, 6(2):418–445, 1996.
- [12] C. J. van Rijsbergen. Information retrieval. 2005. <http://www.dcs.gla.ac.uk/Keith/Preface.html>.
- [13] Cyttron visualization platform. <http://graphics.tudelft.nl/~pkok/CVP>.
- [14] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, V1(4):321–331, January 1988.
- [15] Martin Wildeman. Master thesis: Molecular imaging in bioinformatics, 2007.

CHAPTER 3

Paper on Statistical Testing of Gene Enrichment in *C. Elegans*.

TESTING FOR SPATIAL GENE ENRICHMENT IN *C. ELEGANS* USING CHRONOGRAMS AND A 1D WORM ATLAS.

M. H. Wildeman^{1,2}, M. Baiker², J. H. C. Reiber², B. P. F. Lelieveldt², M. J. T. Reinders¹

¹Information and Communication Theory Group
Department of Mediamatics
Delft Technical University
The Netherlands

²Division of Image Processing
Department of Radiology
Leiden University Medical Center
The Netherlands

ABSTRACT

Based on chronogram data that we obtained from the local-
ize website of Dupuy et al., we explored the possibility
of making qualitative statements on gene expression location,
solely based on the observed gene expression profile regis-
tered to a manually constructed atlas of organ locations. We
show that we are able to make statistically relevant statements
on locally enriched gene expression using this approach and
that we were able to find highly enriched gene expression pro-
files. Different statistical tests are explored, resulting in sim-
ilar top scoring genes for each test. We also show that our
method is limited, because we used a 1D spatial gene expres-
sion profile of 3D organisms, resulting in overlapping quali-
tative spatial locations in our dataset, such as the intestine and
the gonad sheet cells or the pharynx and the head neurons,
resulting in an ambiguous meaning to the found qualitative
expression enrichments.

Index Terms— Gene Enrichment, Chronograms, Statis-
tical Test, Spatial Expression, *Caenorhabditis Elegans*

1. INTRODUCTION

Functional Genomics is a field of science that studies the in-
teractions and functional behavior of genes. For these stud-
ies, large datasets on gene transcription and translation and
protein interactions are used. A common technique used in
genomics to measure expression profiles of mRNA (transcrip-
tion) is the use of microarrays [1]. A major drawback of mi-
croarrays is, that it is difficult and expensive to obtain expres-
sion data in local regions of an organism, because in addition
sectioned tissue profiling is required. If a temporal compo-
nent is also wanted, acquisition of expression becomes a la-
borious task [2]. It is known that gene expression is locally
regulated and therefore spatio-temporal gene expression data
may provide new functional insights on gene regulation. It is
therefore useful to have a means of acquiring spatio-temporal
expression information.

In this study we used spatio-temporal gene expression
data obtained from *C. Elegans*. *C. Elegans* is a well stud-

ied organism in both anatomical and genetical perspective.
Dupuy et al. [3] have acquired images of 1,992 different *C.*
Elegans strains, by making use of Fluorescence Microscopy
and a COPAS sorter (Union Biometrica). The COPAS device
is a flow cytometer that allows to measure objects of 20 to
1500 microns in size. Each strain has a promoter::GFP con-
struct inserted in its genome, which allowed Dupuy et al. to
observe spatial expression profiles of gene expression in a
single worm, by making use of the biofluorescent nature of
the GFP protein¹. Dupuy et al. used the size of worms to
determine the developmental stage of *C. Elegans*. By sorting
all measured worms in size, a temporal component was added
to the dataset, resulting into spatio-temporal gene expression
profiles, which were visualized in what was introduced as
chronograms (Fig 2). To be able to improve our understand-
ing of local gene regulation in organisms, we explored the
possibility of making statistically relevant, qualitative state-
ments on spatially localized gene expression, by comparing
spatial expression profiles, observed in *C. Elegans*, with a
manually constructed 1D atlas of organ locations. First, we
filtered and registered our dataset into a consistent set of
chronograms. Second, we annotated the atlas with organ
locations. Finally, we applied different statistical tests, to de-
termine the presence or absence of enrichment in the organs
locations which were predefined in the atlas.

We show that we are able to find highly enriched gene ex-
pression profiles, based on a dataset of spatio-temporal gene
expression profiles and the use of prior knowledge in the form
of an atlas.

2. METHODS

2.1. Construction of Worm Atlas

To be able to test for enrichment of gene expression in cer-
tain organs, we created an atlas of the organ locations of *C.*
Elegans. To generate this atlas, we gathered information on

¹A promoter::GFP construct is a strand of DNA, containing a promoter
pattern that regulates the transcription rate of the 'attached' gene coding
DNA, in this case the gene for the GFP protein.

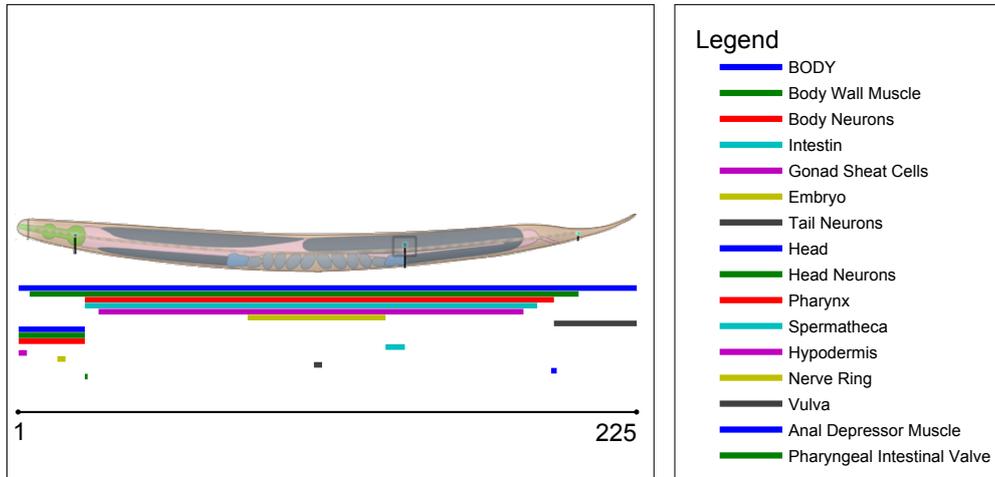


Fig. 1. Manually Constructed Atlas (colored lines), for a reference of organ location and spatial extend, based on a 2D atlas from [4]

organ locations from WormBase [5], WormAtlas [4] and other sources [6]. From the WormAtlas website, we obtained a schematic image which we used to measure locations and sizes of a set of organs. The result of these measurements is shown in Figure 1. Since the obtained spatial expression profiles only had one dimension, we reduced the 2D worm atlas to a 1D projection. Since this reduction in dimensionality causes many organs to overlap, no unambiguous qualitative statements can be made on gene expression location anymore.

2.2. Construction of dataset

Acquisition

We obtained the gene expression dataset, generated by Dupuy et al. from the *C. Elegans* Localizome website [7]. Some of the acquired strains contained the same promoter::GFP construct, so we merged all expression profiles having identical promoters, by averaging the multiple measurements into a new expression profile. This way we acquired a total of 1681 chronograms, each having a unique promoter.

Signal Filtering

To obtain a good quality dataset, we removed all chronograms missing more than 20 percent of data points, caused by non observed worm sizes. A data point is defined as a pixel of a worm. As an effect, more small worm sizes are allowed to be absent than large worms sizes. Because Dupuy et al. stopped their acquisition when most of the worms between size 50 pixels (0.1mm) and 450 pixels ($\approx 1mm$) were observed in their measurements, we decided to discard worm sizes above and below these set boundaries. After removal of poor data sets and non relevant worm sizes, we had a remaining data set of 1155 chronograms, each chronogram having a

size of 401x450 pixels (See Fig. 2A). Worms being smaller than 450 pixels, were aligned to the left side of the chronogram and remaining pixels were set to zero. The data gaps in the remaining chronograms were filled by making use of interpolation (See Fig. 2B), so that we obtained a 'complete' dataset to run our enrichment tests on. Interpolation was done by taking the expression average of the two nearest existing worm sizes smaller in size than the missing worm and the two nearest worm sizes larger in size.

Registration

To be able to compare all worm sizes using the same statistical test, we 'registered' the worms to our manually created worm atlas, by resizing each individual worm to a length of 225 pixels. The size of 225 was chosen arbitrarily, although, keep in mind that we wanted to reduce the weight of the largest worms and increase the weight of the smaller worms in the total dataset. A rationale for this value is, that a small worm keeps more explanatory power, when it is not upscaled too much. Up and downsampling of the worm sizes was done using linear interpolation. We could use a linear registration since *C. Elegans* shows a linear growth during each phase of its development [8]. Inspection of our registered dataset suggests that growth is indeed linear in each phase, since straight lines in expression peaks can be observed in the registered chronograms (See Fig. 2C and Supplementary Figs. 1S - 4S).

Normalization

To be able to compare all promoter expression profiles with each other, we normalized the data per worm. Normalization of the expression curve was done in order to obtain an expression profile with a total expression of 1 (Equation 1). The normalization was applied to make visualization and thus visual comparison of the chronograms easier (See Fig.

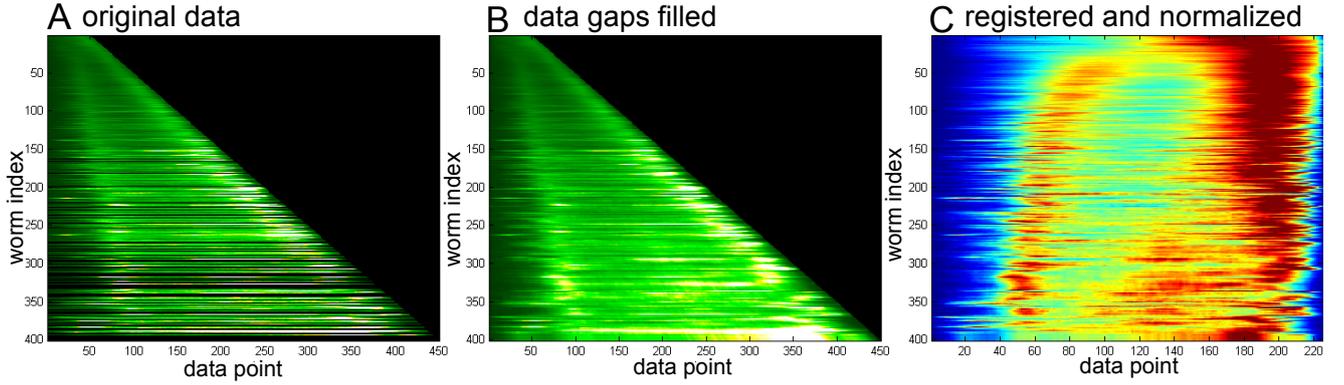


Fig. 2. Chronogram data visualization of a randomly selected promoter. From left to right: A) Original Data, B) Original Data with filled gaps, C) Registered and Normalized Chronogram.

2C and Supplementary Figs. 1S - 4S).

$$\hat{I}_x = \frac{I_x - P_5(I)}{\sum_{x=1}^N I_x - P_5(I)N} \quad (1)$$

In this equation, \hat{I} is the set of normalized intensity values, I the set of original intensity values, N the number of data points in the worm, i.e. its size and x a pixel of a worm. $P_5(I)$ denotes the 5th percentile of set I . Note that N is fixed to 225 after registration.

2.3. Data Integrity

The dataset of Dupuy et al. contains worms that were aligned head to tail automatically, based on their expression profiles. We observed some wrongly aligned worms in some chronograms, suggesting that a Pearson Cross Correlation, which Dupuy et al. used for alignment of worms, is not successful in all cases. It could well be that because of these errors we will miss some highly enriched signals. On the other hand, given the dataset that we use, one could argue that a wrongly aligned chronogram is not enriched. Also the main head/tail orientation of the chronograms was determined manually by Dupuy et al. We assume that this was done correctly, and thus that in all our normalized chronograms, the heads of the worms are located at the left side of the chronograms. We decided to use the dataset of Dupuy et al. as is.

2.4. Qualitative Analysis on Spatial Expression

With a constructed Atlas as reference model and a normalized and registered dataset, we can test for gene expression enrichment, i.e. whether the expression of a gene in a certain organ is significantly higher than outside that organ. Since the constructed atlas is only applicable to the adult stage of *C. Elegans*, we selected only the adult worm sizes in our chronograms, being 325 to 450 pixels in size, roughly corresponding to 650 to 900 μm in unregistered expression profiles. The 126

largest worms were thus used for statistical testing. We first tested whether the expression data was normally distributed, to decide which statistical tests to use. Since no clear normal distribution was observed, we compared parametric and non-parametric tests. We tested for enrichment with the following tests. First, a two-sided two sample t-test, assuming normal distribution, testing for $H_0 : \mu_1 = \mu_2$, where μ is the mean of the corresponding sample. Second, a two-sided Wilcoxon Rank-sum test, to test against $H_0 : median_1 = median_2$, assuming unknown distribution. Lastly we calculated a correlation score between the observed expression profile and an 'expected' expression profile; we assigned a value of -1 to all pixels falling outside of the organ and a value of 1 for all pixels falling inside of it.

Results for the t-test and correlation are expected to be similar, because of Supplementary Equation 9S. A high t-test statistic, Rank-sum (Wilcoxon) or correlation score implicates a high enrichment at the assumed organ location. The choice of tests was driven by [9].

Separate Worm Profiles

First we apply a statistical test to each worm in a chronogram individually, resulting in 126 test statistics for each chronogram. This will result in a matrix of 1155x126x16 in size, containing the values of 1155 chronograms, each having 126 worm sizes tested against 16 different organs. Another matrix having the same size is obtained, containing the p-values corresponding to the statistical tests. For all tests we calculate a *mean* of the test statistic over the worm sizes, yielding a matrix of 1155x16 in size containing test statistics. For the p-values we also calculate the *mean* value over all worm sizes. We can use the test statistics to rank the chronograms on expression enrichment, resulting in 16 ranked lists of chronograms, one for each organ.

Combined Worm Profiles

Instead of calculating the test statistics individually for

each worm size, we can also apply a statistical test to all worm sizes at once, since we registered all worms to our atlas. Combined testing results in one p-value and one test statistic for each chronogram that we can use directly as enrichment scores. Again we obtain a matrix with size 1155x16 containing test statistics and p-values, to calculate a ranking and a significance filter. While with the separate worm size testing we could use the original data, we now have to normalize the data, because we combine different expression profiles. We normalized each worm size in a chronogram separately using Equation 1

Correcting for multiple testing

We can filter out only significant enrichment by making use of the p-values obtained with the statistical tests. To prevent marking enrichments as significant, that are solely caused by chance, we have to correct our p-value cut-off, α for multiple testing [10], and we decided to use a Bonferroni correction for this. Our p-value cut-off of 0.05 is corrected to $\alpha = 0.05/n$ where n in our case is 1155 (# of strains), corresponding to the number of filtered chronograms in our test set.

2.5. Definition Problems

It is not straightforward to state when expression is enriched or not. The minimal definition would be that a signal is said to be enriched in a certain area when the mean expression levels are higher in that area than outside of that area, and have a p-value lower than an α -cutoff of 0.05 corrected with Bonferroni. Any more specific statements are driven by arbitrary decisions.

It could be argued that some minimal difference between two means is needed, to define something as enriched. When, for example, you only define a signal as enriched when it has at least a threefold higher expression than the observed background signal [11], a one tailed two sample t-test could be applied, testing against the null hypothesis $H_0 : \mu_1 = 3\mu_2$. The scalar 3 however, is arbitrary. It is also possible to lower the allowed p-value, to only get really significant enrichments, i.e. having a large difference in mean and a relatively small variance, but setting the right value for the p-value cut-off is also subjective.

We decided to use the minimal definition of enrichment, and tested for significant differences in mean expression signal.

2.6. Validation

For most promoter chronograms we have the availability of expert annotation on qualitative gene expression location, which is based on microscopy observations. To validate our results, we manually counted the occurrence of the searched enrichment versus the corresponding spatial expression tags,

which were present in the localizome database [7]. Each chronogram in the localizome database has a lists of qualitative expression tags attached to it. Labels of chronograms that we merged during the acquisition of the dataset were merged in a new set of tags by making a union of the tag sets.

3. RESULTS

We calculated all test statistics on all chronograms, in combination with all organs defined in our worm atlas for separate worm sizes and combined worm sizes, resulting in two matrices of 1155x126x16 and 1155x16 in size respectively. We calculated the top scoring enrichments for all organs where some organs showed more profound enrichments than others. Figure 3 shows an example of the top 10 enriched promoters for the head, that were found by using the Wilcoxon Rank-sum test and combined worm sizes.

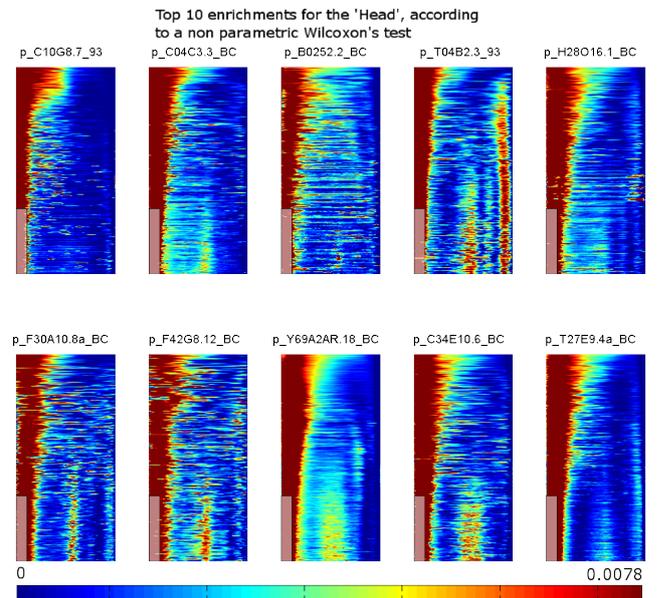


Fig. 3. Result of the top 10 mean sum of the ranks of the Wilcoxon test. The labels shown, are the names of the corresponding promoters. The light overlaid boxes illustrate the tested organ region.

In Table 1 the top 10 scoring promoters for the tail neurons are shown, including their corresponding p-values. Note that ordering the test p-values does not yield the same result as ordering the test statistics, since, because we used two tailed tests and correlation, the p-value only indicates the significance of the found difference in mean and not the direction of difference, e.g. greater or less than the mean of the other sample. Another reason is, that since we calculated the means of the test statistics and the p-values, the ordering of both values changed. This is depicted in Figure 4 (Individual) and Figure 5 (Combined).

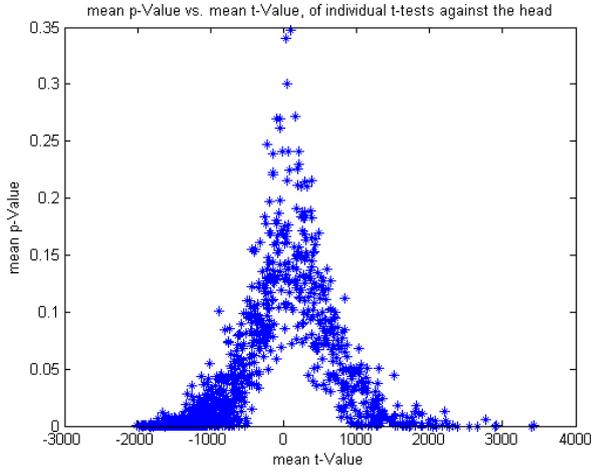


Fig. 4. means of t-Values and p-Values, for t-test against the head. Wormsizes were tested individually, and resulting p-Values and t-Values were put in a new mean value. The mean p-Values are not fully determined by mean t-Values anymore, but are related.

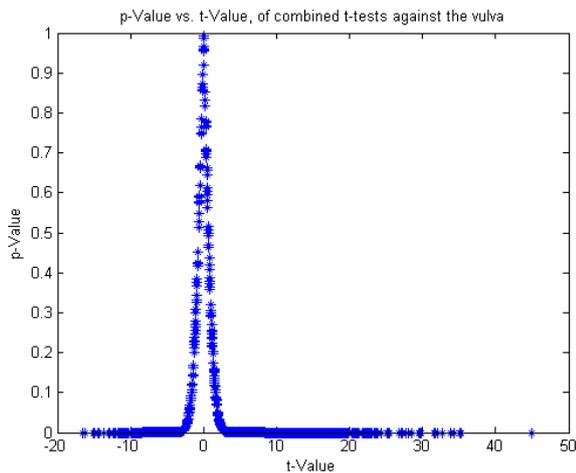


Fig. 5. p-Values versus t-Test statistics, for t-tests against the vulva. Wormsizes were combined in this test and p-Values are fully determined by t-Values.

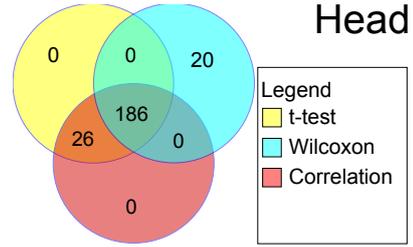


Fig. 6. Overlap of found significant difference in mean, for the tested expression profiles against the head, for different statistical tests.

We could have filtered out the test statistics that are below the expected value; 0 for correlation and t-test and, $113 * \text{organize}$ for the Wilcoxon rank sum test (see Equation 15S for derivation) and allow a p-value cut-off that is twice as large, but we think it is interesting to keep both tails in our set, because it yields the enriched and not enriched examples (red samples in Table 1). When we tested on the combined worm sizes, we obtained very small p-values, which were shown as 0 in Matlab. We could therefore not use p-values to rank the found enrichment in significance. Since in the combined worm profiles tests, the relation between p-values and test statistics was maintained (Figure 5), we could in this case safely order our enrichments, based on the test statistics instead of the p-values (See Tables 2S and 3S). We compared the found results of the t-test, the Wilcoxon rank-sum test and a correlation coefficient. The correlation and t-test statistics show the same behavior, as expected. A Venn diagram is shown in Figure 6 to illustrate this. Since all tests perform similar, it is preferred to make use of the Wilcoxon's Rank-sum test for enrichment tests, because this test does not make any assumptions on the underlying distributions of the observed expression data.

4. DISCUSSION

We are able to find highly enriched signal with the use of a manually constructed atlas. We showed that combining all worm sizes into one test gives more significant enrichments than when we test each worm size separately. The results indicate that a manually constructed atlas can aid in finding areas with enriched gene expression and that found enrichments show a common pattern. Also correct spatial expression tags were found in many cases. The atlas that we use is not fully annotated because many organs show overlap and full annotation would give much redundancy. We do have some remaining problems that need to be solved.

4.1. Limitations

Since many organs overlap in our 1D atlas, the exact recovery rate is low. For example when searching for enrichment in

		Statistical Test on individual worm sizes									
		t-test		wilcoxon				correlation			overlap
organ	promoter	mean t-value	mean p-value	promoter	mean ranksum	mean p-value	promoter	mean R	mean p-value	Venn	
Tail Neurons	p-W08E12.5.93	2232	1.54e-012	p-W08E12.5.93	5908	5.86e-007	p-W08E12.5.93	0.75	1.54e-012		
	p-D1007.1.93	1748	1.46e-006	p-Y16B4A.1.93	5790	2.51e-007	p-D1007.1.93	0.66	1.46e-006		
	p-Y16B4A.1.93	1713	2.07e-010	p-F42H10.3.93	5674	8.01e-006	p-Y16B4A.1.93	0.66	2.07e-010		
	p-C45G9.13_BC	1611	1.81e-007	p_F09C3.2.93	5592	7.9e-006	p_C45G9.13_BC	0.63	1.81e-007		
	p_F09C3.2.93	1400	1.61e-010	p_D1007.1.93	5569	1.25e-005	p_F09C3.2.93	0.59	1.61e-010		
	p_C12D8.1a_BC	-704	3.93e-005	p_F19C6.1_BC	5448	3.13e-005	p_C12D8.1a_BC	-0.35	3.93e-005		
	p_Y69A2AR.18_BC	-738	2.92e-007	p_T25B9.2_BC	1365	1.77e-005	p_Y69A2AR.18_BC	-0.36	2.92e-007		
	p_C18A3.5a_BC	-764	2.65e-005	p_F52E4.1a.93	1309	3.61e-005	p_C18A3.5a_BC	-0.37	2.65e-005		
	p_Y37A1B.5_BC	-784	3.79e-005	p_W09D6.6_BC	1207	1.93e-005	p_Y37A1B.5_BC	-0.38	3.79e-005		
	p_C41D11.2_BC	-820	1.94e-005	p_K07D4.7b_BC	1116	7.41e-006	p_C41D11.2_BC	-0.4	1.94e-005		

Table 1. Top scoring enrichments for different tests, using separate worm sizes. For full result table see Supplementary Tables 1S,2S. For visualization see Supplementary Figures 1S - 4S. Red values mean that there is a significant difference in the sample value, but the mean expression in the organ region is smaller than outside of it (depleted).

the nerve ring, also promoters being expressed in the head neurons, pharynx, and head show up, and vice versa.

We observed that many promoters did not have unambiguous expression tags associated with them, because different strains belonging to the same promoters contained different expression annotation tags. The list of anatomical annotation in the database was also not complete, only having 72% of all promoters annotated, suggesting that the existing set of annotations in the database is not complete either.

When ignoring the missing annotations, and using a (subjective) region for validation, instead of the specific organ, results were still difficult to interpret, but some correct annotations were found. I.e. all tags of organs that are (partly) located in the region of the organ that is tested are seen as correct enrichment hit, or true positive. No accurate validation could thus be done, and so the real performance of the used tests remains unknown. Next to a list of correct tags, also many unrelated tags show up, which is not surprising. The fact that we do detect enrichment in the region of the head, does not mean that the gene is not expressed in other parts of the body, only that it is expressed at a different (detectable) level. Resulting annotations from top scoring promoters can be found in the supplementary (Table 4S).

4.2. Atlas Registration

We have no guarantee that the constructed atlas is accurate, nor have we the knowledge that each worm has the same proportions and organ locations. It could well be that the growth of organisms is disturbed by the genomic alteration, resulting from changes in protein synthesis and anabolic requirements for this. As mentioned earlier, since we have overlapping organs in our atlas, it is not possible to make unambiguous qualitative statements on gene expression. This clearly shows the limits of the approach we used and forces us to maintain a loose definition of enrichment to be in areas instead of in organs.

5. FUTURE WORK

We want to extend this research from *C.Elegans* to expression profiles of a normal lab mouse. There is a structural

atlas available in the form of Digimouse [12], that we can use for making statements on spatial expression in a similar way as we did in this work. To extend this work to a 3D model, instead of the current 1D model, some modifications have to be made and some cautions should be taken. In a 3D atlas approach, we no longer have overlapping organs, as we did with the *C. Elegans* atlas. This will make the qualitative statements that we can make, less ambiguous. Since mouse models are often used as model organisms for cancer research, an example for enrichment search would be, that we want to answer the question whether a cancer cell line has metastasized to the bone or not; we only expect tumor growth on parts of the skeleton, and not on the complete skeleton. We can use the atlas to separate the body of the mouse into two segments, e.g. 'skeleton' and 'not skeleton', resulting into two labeled subsets of expression data which we can compare in the same way as we did with this research. This example suggests that this work can almost directly be translated to other organisms, as long as we have the availability of a structural anatomical atlas that can be registered to the acquired expression profiles.

Expected Problems

Note that in order to obtain a 3D expression dataset of a mouse model, we will need a good atlas registration of both skin surface and internal organs, in order to be able to reconstruct the source of fluorescence by making use of a photon propagation model [13]. If we are not able to locate the source of fluorescence, we will not have the availability of a 3D dataset and will have to use a 2D expression profile, resulting in overlapping organs again. We expect that 3D organs will have localized expression internally, which will make it more difficult to obtain statistical relevant statements on enrichment, because enrichment in a small part of an organ would not significantly increase the mean expression compared to the rest of the mouse body. The metastasis example mentioned before, clearly shows this.

6. REFERENCES

- [1] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown. Quantitative monitoring of gene expression pat-

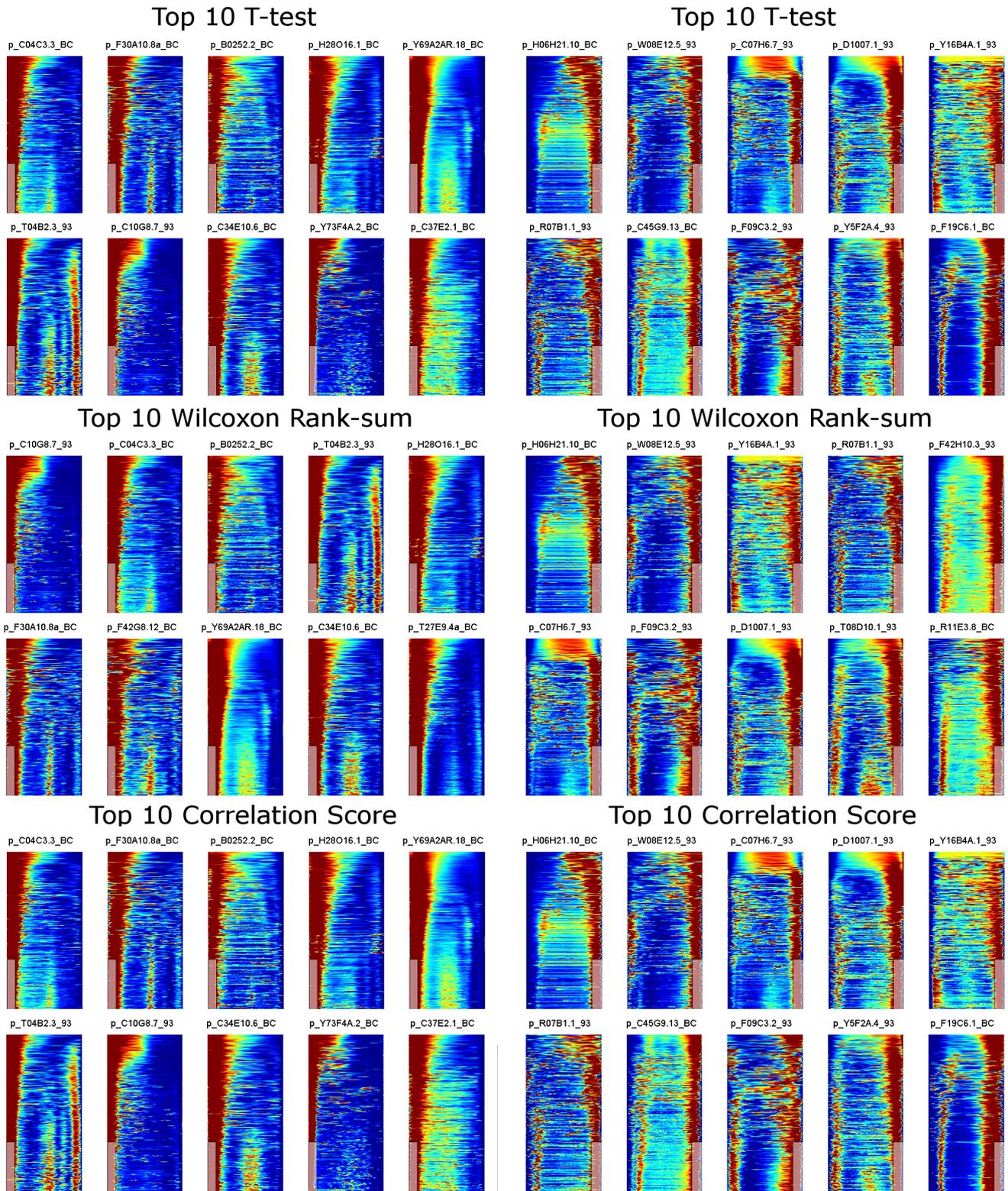
- terns with a complementary dna microarray. *Science*, 270(5235):467–470, October 1995.
- [2] H. Peng, F. Long, J. Zhou, G. Leung, M.B. Eisen, and E.W. Myers. Automatic image analysis for gene expression patterns of fly embryos. *BMC Cell Biology*, 8, July 2007.
- [3] D. Dupuy, N. Bertin, C.A. Hidalgo, K. Venkatesan, D. Tu, D. Lee, J. Rosenberg, N. Svrikapa, A. Blanc, A. Carnac, A. Carvunis, R. Pulak, J. Shingles, J. Reece-Hoyes, R. Hunt-Newbury, R. Viveiros, W.A. Mohler, M. Tasa, F. P. Roth, C. Le Peuch, I.A. Hope, R. Johnsen, D.G. Merman, A. L. Barbasi, D. Baillie, and M. Vidal. Genome-scale analysis of *in vivo* spatiotemporal promoter activity in *Caenorhabditis elegans*. *Nature Biotechnology*, 25(6):663–668, June 2007.
- [4] Z. F. Altun and D. H. Hall. Handbook of *c. elegans* anatomy. *WormAtlas*, 2005. <http://www.wormatlas.org/handbook/contents.htm>.
- [5] Wormbase web site, 2008. <http://www.wormbase.org>, release WS196.
- [6] R. Fox, J. Watson, S. Von Stetina, J. McDermott, T. Brodigan, T. Fukushige, M. Krause, and D. Miller. The embryonic muscle transcriptome of *caenorhabditis elegans*. *Genome Biology*, 8(9):R188, 2007.
- [7] D. Dupuy. *C. Elegans* localizome project. <http://localizome.dfci.harvard.edu>.
- [8] C. G. Knight, M. N. Patel, A. M. Leroi, and R. B. Azevedo. A novel mode of ecdysozoan growth in *caenorhabditis*. *Evolution & Development*, 4:16–27, 2002.
- [9] Motulsky H. *Intuitive Biostatistics*. New York: Oxford University Press, 1st edition, 1995.
- [10] N.J. Salkind. *Encyclopedia of Measurement and Statistics*. Thousand Oaks (CA): Sage.
- [11] G. Gu, J. M. Wells, D. Dombkowski, F. Preffer, B. Aronow, and Douglas A. Melton. Global expression analysis of gene regulatory pathways during endocrine pancreatic development. *Development*, 131(1):165–179, 2004.
- [12] B. Dogdas, D. Stout, A. Chatziioannou, and R. M. Leahy. Digimouse: A 3d whole body mouse atlas from ct and cryosection data. *Phys Med Biol.*, 52:577–87, February 2007.
- [13] M Wildeman. Molecular imaging in bioinformatics. Research Assignment, February 2008.

**TESTING FOR SPATIAL GENE ENRICHMENT IN *C. ELEGANS* CHRONOGRAMS.
- SUPPLEMENTARY MATERIALS -**

M. H. Wildeman^{1,2}, *M. Baiker*², *J. H. C. Reiber*², *B. P. F. Lelieveldt*², *M. J. T. Reinders*¹

¹Information and Communication Theory Group
Department of Mediamatics
Delft Technical University
The Netherlands

²Division of Image Processing
Department of Radiology
Leiden University Medical Center
The Netherlands



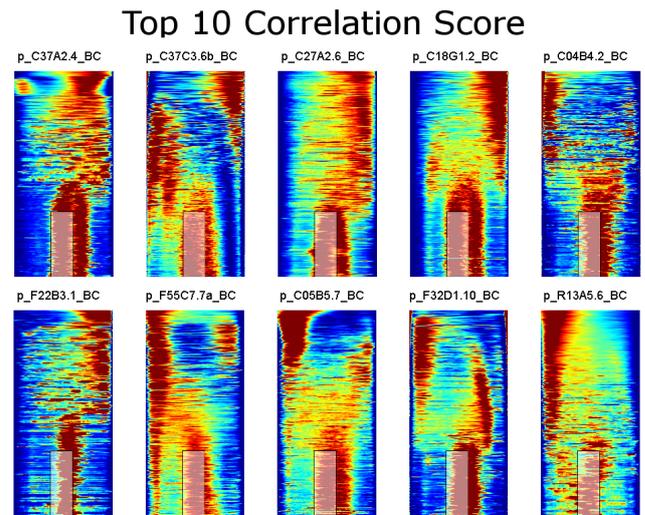
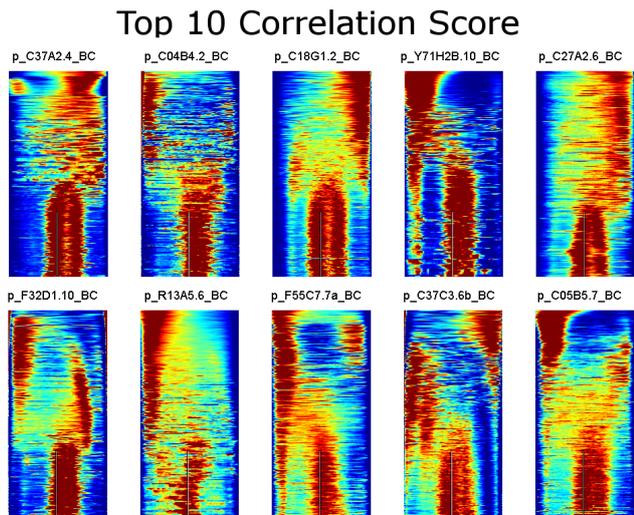
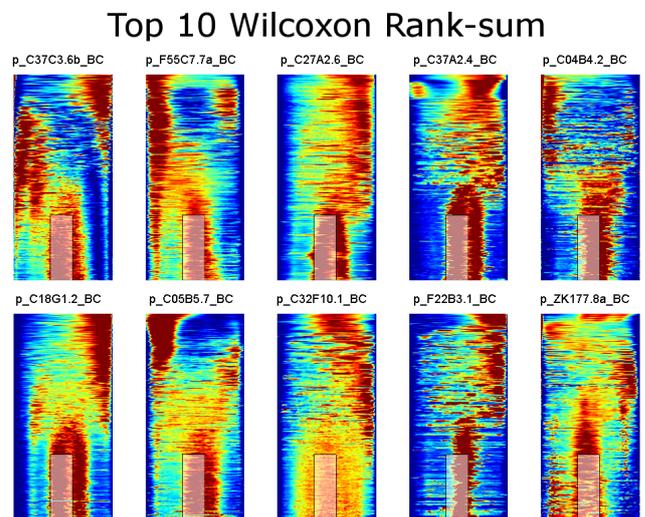
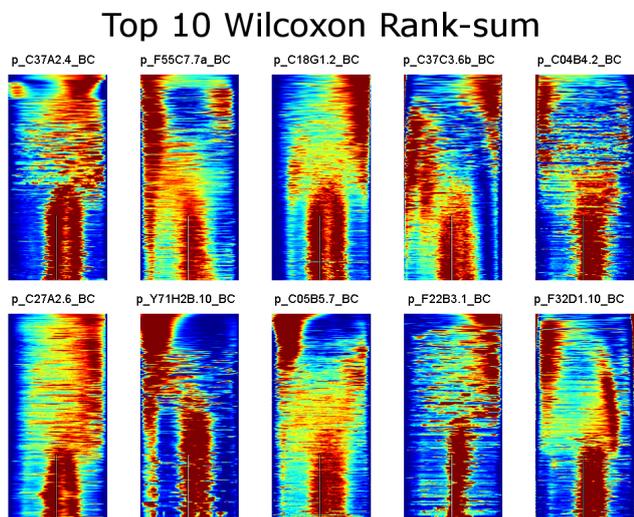
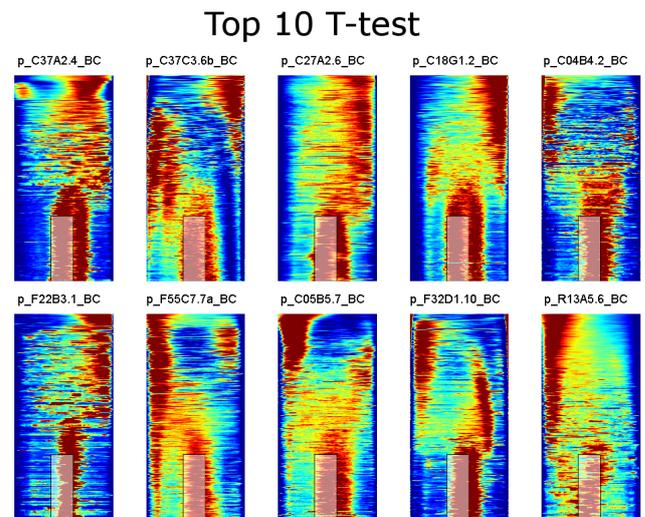
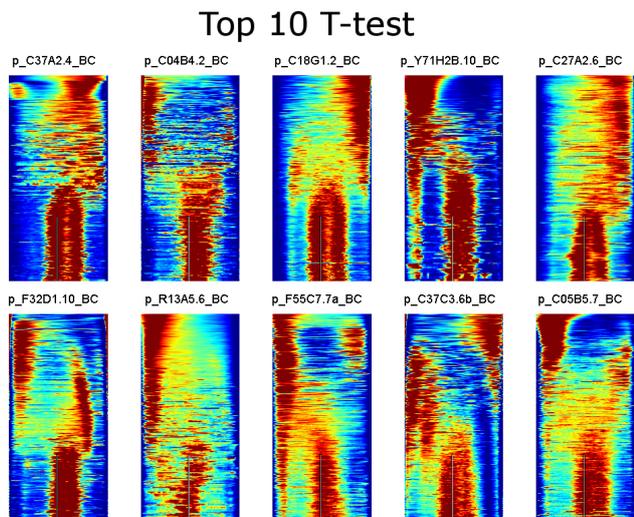


Fig. 3S: Top 10 Scoring Enrichments for the Vulva. Tested area depicted as rectangle overlay.

Fig. 4S: Top 10 Scoring Enrichments for the Developing Embryo. Tested area depicted as rectangle overlay.

organ	Statistical Tests on Registered Data, Individual Worms, Top 10 Scores										Venn
	t-test		wilcoxon				correlation		Overlap		
	promoter	mean t-value	mean p-value	promoter	mean ranksum	mean p-value	promoter	mean R	mean p-value		
Body Wall Muscle	p.C29E4.4_BC p.F55A8.2b_BC p.F52F12.1a_BC p.B0228.4_BC p.F39G3.7_BC p.K07G5.3_BC p.AC7.2_BC p.C16C10.1_BC p.C53D6.2_BC p.C52G5.2_93	1925 1887 1846 1804 1753 1750 1695 1694 1646 1605	3.15e-016 3.27e-005 1.59e-017 2.35e-009 2.51e-006 1.23e-012 2.57e-008 6.56e-007 2.34e-008 6.24e-012	p.F01G12.5a_BC p.K07G5.3_BC p.F52F12.1a_BC p.F28F8.6_BC p.F39G3.7_BC p.W08F4.3_BC p.Y41D4A.4_BC p.M01E11.7_BC p.Y71G12B.15_BC p.R09F10.6_BC	25036 25033 25025 25016 25010 25010 25010 25009 25007 25006	1.49e-011 6.54e-014 1.17e-012 8.2e-009 1.29e-009 7.57e-010 2.86e-009 9.46e-012 2.31e-005 3.2e-012	p.C29E4.4_BC p.F52F12.1a_BC p.F55A8.2b_BC p.K07G5.3_BC p.B0228.4_BC p.F39G3.7_BC p.AC7.2_BC p.C16C10.1_BC p.C53D6.2_BC p.M01E11.7_BC	0.7 0.69 0.68 0.67 0.67 0.66 0.65 0.65 0.64 0.64	3.16e-016 1.6e-017 3.27e-005 1.23e-012 2.35e-009 2.51e-006 2.57e-008 6.56e-007 2.34e-008 5.01e-016		
Body Neurons	p.K07D4.7b_BC p.K07G5.3_BC p.C47C12.3_BC p.T25B9.2_BC p.C05A9.1_BC p.Y67D2.2_BC p.F32B4.2_93 p.F56B6.4b_BC p.AH6.4_BC p.T10C6.13_BC	3022 2729 2654 2608 2570 2449 2373 2366 2356 2347	1.28e-037 3.59e-005 4.06e-014 2.31e-006 8.99e-007 7.95e-009 2.55e-005 2.89e-010 3.63e-010 8.07e-013	p.K07D4.7b_BC p.C56G2.9_BC p.T10C6.13_BC p.K07G5.3_BC p.C47C12.3_BC p.F54C9.2_BC p.Y67D2.2_BC p.F56B6.4b_BC p.C04B4.2_BC p.ZC373.5_BC	23815 23720 23714 23683 23677 23617 23605 23598 23584 23581	2.77e-019 3.31e-014 5.61e-006 8.91e-006 6.18e-011 2.54e-005 8.58e-007 2.49e-012 1.14e-007 3.76e-007	p.K07D4.7b_BC p.C47C12.3_BC p.K07G5.3_BC p.T25B9.2_BC p.C05A9.1_BC p.Y67D2.2_BC p.C56G2.9_BC p.AH6.4_BC p.T10C6.13_BC p.F56B6.4b_BC	0.84 0.81 0.8 0.8 0.79 0.78 0.77 0.77 0.76 0.76	2.28e-021 4.06e-014 3.59e-005 2.31e-006 8.99e-007 7.95e-009 7.04e-019 3.63e-010 8.07e-013 2.89e-010		
Intestin	p.K07G5.3_BC p.C01G8.5a_BC p.K07D4.7b_BC p.AH6.4_BC p.Y45G12C.10_BC p.F32B4.2_93 p.C47C12.3_BC p.T25B9.2_BC p.C56G2.9_BC p.T10C6.13_BC	2874 2847 2840 2460 2430 2409 2404 2389 2381 2330	2.31e-006 5.59e-006 1.99e-039 4.3e-008 1.27e-005 2.38e-006 2.65e-010 1.2e-005 1.23e-012 3.91e-012	p.K07D4.7b_BC p.K07G5.3_BC p.T10C6.13_BC p.C56G2.9_BC p.K08E7.8_BC p.F54C9.2_BC p.C01G8.5a_BC p.F11H8.1_BC p.F35H8.5_BC p.ZC373.5_BC	23345 23344 23309 23299 23286 23275 23232 23223 23222 23216	1.22e-017 2.2e-007 2.99e-006 7.04e-010 1.17e-005 7.21e-007 3.61e-007 3.52e-006 1.7e-006 1.7e-008	p.K07D4.7b_BC p.K07G5.3_BC p.C01G8.5a_BC p.AH6.4_BC p.C56G2.9_BC p.C47C12.3_BC p.T25B9.2_BC p.Y45G12C.10_BC p.T10C6.13_BC p.F32B4.2_93	0.82 0.82 0.8 0.78 0.78 0.77 0.77 0.77 0.76 0.76	4.69e-022 2.31e-006 5.59e-006 4.3e-008 1.23e-012 2.65e-010 1.2e-005 1.27e-005 3.91e-012 2.38e-006		
Gonad Sheat Cells	p.K07G5.3_BC p.K07D4.7b_BC p.C56G2.9_BC p.AH6.4_BC p.T10C6.13_BC p.C47C12.3_BC p.Y67D2.2_BC p.C05A9.1_BC p.T25B9.2_BC p.ZC373.5_BC	2954 2800 2671 2632 2459 2425 2382 2348 2286 2266	2.05e-006 3.32e-015 5.4e-011 9.21e-010 4.37e-013 1.23e-007 2.24e-006 1.92e-006 2.36e-006 3.58e-008	p.C56G2.9_BC p.K07G5.3_BC p.T10C6.13_BC p.K08E7.8_BC p.F35H8.5_BC p.F54C9.2_BC p.F11H8.1_BC p.K07D4.7b_BC p.ZC373.5_BC p.C04B4.2_BC	22683 22641 22615 22605 22589 22572 22549 22508 22488 22473	8.37e-009 2.2e-007 5.32e-008 3.59e-005 2.08e-008 6.04e-007 5.24e-006 4.57e-015 1.91e-007 6.65e-010	p.K07G5.3_BC p.K07D4.7b_BC p.C56G2.9_BC p.AH6.4_BC p.T10C6.13_BC p.C47C12.3_BC p.Y67D2.2_BC p.C05A9.1_BC p.T25B9.2_BC p.ZC373.5_BC	0.82 0.82 0.81 0.8 0.78 0.77 0.77 0.76 0.75 0.75	2.05e-006 3.32e-015 5.4e-011 9.21e-010 4.37e-013 1.23e-007 2.24e-006 1.92e-006 2.36e-006 3.58e-008		
Embryo	p.C37A2.4_BC p.C18G1.2_BC p.F55C7.7a_BC p.C32F10.1_BC	1597 1481 1371 912	1.5e-011 2.57e-005 7.39e-008 8.31e-007	p.F55C7.7a_BC p.C37A2.4_BC p.F22B3.1_BC p.C04B4.2_BC p.C32F10.1_BC p.C26D10.2_BC	9477 9360 9313 9280 9232 8719	2.64e-008 8.59e-012 1.72e-009 1.27e-008 1.07e-006 1.47e-005	p.C37A2.4_BC p.C18G1.2_BC p.F55C7.7a_BC p.C32F10.1_BC	0.64 0.61 0.58 0.43	1.5e-011 2.57e-005 7.39e-008 8.31e-007		
Tail Neurons	p.W08E12.5_93 p.D1007.1_93 p.Y16B4A.1_93 p.C45G9.13_BC p.F09C3.2_93 p.C12D8.1a_BC p.Y69A2AR.18_BC p.C18A3.5a_BC p.Y37A1B.5_BC p.C41D11.2_BC	2232 1748 1713 1611 1400 -704 -738 -764 -784 -820	1.54e-012 1.46e-006 2.07e-010 1.81e-007 1.61e-010 3.93e-005 2.92e-007 2.65e-005 3.79e-005 1.94e-005	p.W08E12.5_93 p.Y16B4A.1_93 p.F42H10.3_93 p.F09C3.2_93 p.D1007.1_93 p.F19C6.1_BC p.T25B9.2_BC p.F52E4.1a_93 p.W09D6.6_BC p.K07D4.7b_BC	5908 5790 5674 5592 5569 5448 1365 1309 1207 1116	5.86e-007 2.51e-007 8.01e-006 7.9e-006 1.25e-005 3.13e-005 1.77e-005 3.61e-005 1.93e-005 7.41e-006	p.W08E12.5_93 p.D1007.1_93 p.Y16B4A.1_93 p.C45G9.13_BC p.F09C3.2_93 p.C12D8.1a_BC p.Y69A2AR.18_BC p.C18A3.5a_BC p.Y37A1B.5_BC p.C41D11.2_BC	0.75 0.66 0.66 0.63 0.59 -0.35 -0.36 -0.37 -0.38 -0.4	1.54e-012 1.46e-006 2.07e-010 1.81e-007 1.61e-010 3.93e-005 2.92e-007 2.65e-005 3.79e-005 1.94e-005		
Head Head Neurons Pharynx	p.C04C3.3_BC p.F30A10.8a_BC p.Y69A2AR.18_BC p.C10G8.7_93 p.C34E10.6_BC p.C37E2.1_BC p.Y73F4A.2_BC p.F42G8.12_BC p.R07G3.8_93 p.C03G5.1_BC	3390 2905 2648 2547 2387 2326 2274 2243 2176 2171	2.39e-043 1.82e-037 1.34e-040 8.43e-039 5.44e-038 1.02e-005 1.4e-037 1.56e-016 7.49e-009 3.72e-013	p.C04C3.3_BC p.C10G8.7_93 p.B0252.2_BC p.H28O16.1_BC p.F30A10.8a_BC p.F42G8.12_BC p.Y69A2AR.18_BC p.C34E10.6_BC p.T27E9.4a_BC p.T27A1.4_BC	5204 5289 5251 5242 5244 5213 5211 5198 5163 5129	1.9e-015 8.13e-014 1.51e-006 3.8e-008 2.17e-011 2.98e-006 1.51e-014 5.68e-011 3.61e-014 3.8e-011	p.C04C3.3_BC p.F30A10.8a_BC p.Y69A2AR.18_BC p.C10G8.7_93 p.C34E10.6_BC p.Y73F4A.2_BC p.C37E2.1_BC p.F42G8.12_BC p.C03G5.1_BC p.T27E9.4a_BC	0.87 0.82 0.81 0.79 0.77 0.76 0.75 0.75 0.75 0.74	2.39e-043 2.59e-023 1.96e-031 8.4e-024 2.68e-020 5.53e-026 1.02e-005 1.56e-016 3.72e-013 8.41e-030		
Spermatheca	None Significant			None Significant			None Significant				
Hypodermis	p.F54D8.3_BC p.W02C12.3b_BC p.gel-13.PF p.ZK484.2_BC p.C40C9.5_BC p.C35D10.14_BC p.F28B12.3_BC p.F13G3.7_BC p.F49E8.3_93 p.T12D8.8_BC	-620 -621 -626 -630 -636 -638 -648 -654 -658 -662	1.47e-005 1.01e-005 1.92e-005 2.58e-005 3.44e-005 7.49e-006 3.97e-005 3.34e-005 3.94e-005 2.89e-005	p.F54D8.3_BC p.W02C12.3b_BC p.gel-13.PF p.ZK484.2_BC p.C40C9.5_BC p.C35D10.14_BC p.F28B12.3_BC p.F13G3.7_BC p.F49E8.3_93 p.T12D8.8_BC	-0.31 -0.31 -0.32 -0.32 -0.32 -0.32 -0.32 -0.33 -0.33 -0.33	1.47e-005 1.01e-005 1.92e-005 2.58e-005 3.44e-005 7.49e-006 3.97e-005 3.34e-005 3.94e-005 2.89e-005	p.F54D8.3_BC p.W02C12.3b_BC p.gel-13.PF p.ZK484.2_BC p.C40C9.5_BC p.C35D10.14_BC p.F28B12.3_BC p.F13G3.7_BC p.F49E8.3_93 p.T12D8.8_BC	-0.31 -0.31 -0.32 -0.32 -0.32 -0.32 -0.32 -0.33 -0.33 -0.33	1.47e-005 1.01e-005 1.92e-005 2.58e-005 3.44e-005 7.49e-006 3.97e-005 3.34e-005 3.94e-005 2.89e-005		
Nerve Ring	p.F30A10.8a_BC p.R05F9.1a_BC p.T04A6.1a_BC p.C34E10.6_BC p.C03G5.1_BC p.F08C6.7_93 p.T27A1.4_BC	840 801 795 707 686 652 651	9.47e-007 1.13e-005 1.42e-005 8.69e-006 8.63e-006 4.72e-006 2.9e-006	None Significant			p.F30A10.8a_BC p.R05F9.1a_BC p.T04A6.1a_BC p.C34E10.6_BC p.C03G5.1_BC p.F08C6.7_93 p.T27A1.4_BC	0.41 0.39 0.39 0.35 0.34 0.33 0.33	9.47e-007 1.13e-005 1.42e-005 8.69e-006 8.63e-006 4.72e-006 2.9e-006		
Vulva	None Significant			None Significant			None Significant				
Anal Depressor Muscle	None Significant			None Significant			None Significant				
Pharyngeal Intestinal Valve	None Significant			None Significant			None Significant				

Table 1S: Top scoring enrichments for different tests, using separate worm sizes.

Statistical Tests on Registered Data, Combined Worms, Top 10 Scores										
organ	t-test			wilcoxon			correlation			Overlap
	promoter	t-value	p-value	promoter	ranknum	p-value	promoter	R	p-value	Venn
Body Wall Muscle	p_C29E4.4_BC	162	0	p_F01G12.5a_BC	39577998	0	p_C29E4.4_BC	0.69	0	
	p_F52F12.1a_BC	157	0	p_K07G5.3_BC	395679627	0	p_F52F12.1a_BC	0.68	0	
	p_K07G5.3_BC	150	0	p_F52F12.1a_BC	395675324	0	p_K07G5.3_BC	0.66	0	
	p_F32B4.2.93	147	0	p_F28F8.6_BC	395611117	0	p_F32B4.2.93	0.66	0	
	p_F55C5.8_BC	144	0	p_Y41D4A.4_BC	395407759	0	p_F55C5.8_BC	0.65	0	
	p_C01G8.5a_BC	143	0	p_F55C5.8_BC	395374173	0	p_C01G8.5a_BC	0.65	0	
	p_B0228.4_BC	143	0	p_Y71G12B.15_BC	395354079	0	p_B0228.4_BC	0.65	0	
	p_F10F2.8_BC	141	0	p_F39G3.7_BC	395345890	0	p_F10F2.8_BC	0.64	0	
	p_F55A8.2b_BC	140	0	p_C29E4.4_BC	395320793	0	p_F55A8.2b_BC	0.64	0	
	p_M01E11.7_BC	137	0	p_R09F10.6_BC	395305068	0	p_M01E11.7_BC	0.63	0	
Body Neurons	p_K07D4.7b_BC	251	0	p_K07D4.7b_BC	376569120	0	p_K07D4.7b_BC	0.83	0	
	p_C47C12.3_BC	224	0	p_C39F7.1_BC	375088942	0	p_C47C12.3_BC	0.8	0	
	p_K07G5.3_BC	222	0	p_T10C6.13_BC	375047629	0	p_K07G5.3_BC	0.8	0	
	p_K01G5.2a_BC	214	0	p_K12C11.2_BC	374955680	0	p_K01G5.2a_BC	0.79	0	
	p_C05A9.1_BC	212	0	p_C56G2.9_BC	374892706	0	p_C05A9.1_BC	0.78	0	
	p_C26C6.5_BC	212	0	p_C47C12.3_BC	374853464	0	p_C26C6.5_BC	0.78	0	
	p_T25B9.2_BC	209	0	p_AH6.4_BC	374533045	0	p_T25B9.2_BC	0.78	0	
	p_Y45G12C.10_BC	206	0	p_K07G5.3_BC	374520855	0	p_Y45G12C.10_BC	0.77	0	
	p_C01G8.5a_BC	204	0	p_T25B9.2_BC	373912339	0	p_C01G8.5a_BC	0.77	0	
	p_C56G2.9_BC	202	0	p_K08E7.8_BC	373402719	0	p_C56G2.9_BC	0.77	0	
Intestin	p_K07D4.7b_BC	238	0	p_C39F7.1_BC	369399033	0	p_K07D4.7b_BC	0.82	0	
	p_K07G5.3_BC	233	0	p_K07G5.3_BC	369171865	0	p_K07G5.3_BC	0.81	0	
	p_C01G8.5a_BC	216	0	p_K07D4.7b_BC	369099341	0	p_C01G8.5a_BC	0.79	0	
	p_K01G5.2a_BC	211	0	p_AH6.4_BC	368960234	0	p_K01G5.2a_BC	0.78	0	
	p_AH6.4_BC	209	0	p_T10C6.13_BC	368581266	0	p_AH6.4_BC	0.78	0	
	p_C56G2.9_BC	204	0	p_C56G2.9_BC	368255082	0	p_C56G2.9_BC	0.77	0	
	p_C47C12.3_BC	203	0	p_K08E7.8_BC	368118891	0	p_C47C12.3_BC	0.77	0	
	p_C16A3.10a_BC	202	0	p_K12C11.2_BC	367965459	0	p_C16A3.10a_BC	0.77	0	
	p_T21H3.1_BC	199	0	p_F54C9.2_BC	367867779	0	p_T21H3.1_BC	0.76	0	
	p_Y45G12C.10_BC	199	0	p_C01G8.5a_BC	367527618	0	p_Y45G12C.10_BC	0.76	0	
Gonad Sheat Cells	p_K07G5.3_BC	237	0	p_AH6.4_BC	358864242	0	p_K07G5.3_BC	0.82	0	
	p_K07D4.7b_BC	236	0	p_C56G2.9_BC	358608734	0	p_K07D4.7b_BC	0.81	0	
	p_C56G2.9_BC	229	0	p_C39F7.1_BC	358581749	0	p_C56G2.9_BC	0.81	0	
	p_K01G5.2a_BC	227	0	p_K07G5.3_BC	358240873	0	p_K01G5.2a_BC	0.8	0	
	p_AH6.4_BC	222	0	p_T10C6.13_BC	357834024	0	p_AH6.4_BC	0.8	0	
	p_F56E10.4_BC	214	0	p_F35H8.5_BC	357520473	0	p_F56E10.4_BC	0.79	0	
	p_Y45G12C.10_BC	213	0	p_F56E10.4_BC	357500595	0	p_Y45G12C.10_BC	0.78	0	
	p_Y71G12B.16_BC	206	0	p_K08E7.8_BC	357443329	0	p_Y71G12B.16_BC	0.77	0	
	p_C01G8.5a_BC	205	0	p_K12C11.2_BC	357095947	0	p_C01G8.5a_BC	0.77	0	
	p_C47C12.3_BC	204	0	p_K01G5.2a_BC	356947279	0	p_C47C12.3_BC	0.77	0	
Embryo	p_C37A2.4_BC	140	0	p_C37C3.6b_BC	153503806	0	p_C37A2.4_BC	0.64	0	
	p_C37C3.6b_BC	137	0	p_F55C7.7a_BC	149462722	0	p_C37C3.6b_BC	0.63	0	
	p_C27A2.6_BC	132	0	p_C27A2.6_BC	148421736	0	p_C27A2.6_BC	0.62	0	
	p_C18G1.2_BC	129	0	p_C37A2.4_BC	148077133	0	p_C18G1.2_BC	0.61	0	
	p_C04B4.2_BC	127	0	p_C04B4.2_BC	146738088	0	p_C04B4.2_BC	0.6	0	
	p_F22B3.1_BC	120	0	p_C18G1.2_BC	146613474	0	p_F22B3.1_BC	0.58	0	
	p_F55C7.7a_BC	119	0	p_C05B5.7_BC	146194897	0	p_F55C7.7a_BC	0.58	0	
	p_C05B5.7_BC	118	0	p_C32F10.1_BC	145119286	0	p_C05B5.7_BC	0.57	0	
	p_F32D1.10_BC	114	0	p_F22B3.1_BC	144880459	0	p_F32D1.10_BC	0.56	0	
	p_R13A5.6_BC	110	0	p_ZK177.8a_BC	143903062	0	p_R13A5.6_BC	0.55	0	
Tail Neurons	p_H06H21.10_BC	200	0	p_H06H21.10_BC	97092455	0	p_H06H21.10_BC	0.76	0	
	p_W08E12.5.93	187	0	p_W08E12.5.93	93689865	0	p_W08E12.5.93	0.74	0	
	p_C07H6.7.93	152	0	p_Y16B4A.1.93	91725415	0	p_C07H6.7.93	0.67	0	
	p_D1007.1.93	146	0	p_R07B1.1.93	91369496	0	p_D1007.1.93	0.65	0	
	p_Y16B4A.1.93	141	0	p_F42H10.3.93	90053228	0	p_Y16B4A.1.93	0.64	0	
	p_R07B1.1.93	138	0	p_C07H6.7.93	89425684	0	p_R07B1.1.93	0.63	0	
	p_C45G9.13_BC	135	0	p_F09C3.2.93	88479834	0	p_C45G9.13_BC	0.62	0	
	p_F09C3.2.93	122	0	p_D1007.1.93	88391876	0	p_F09C3.2.93	0.59	0	
	p_Y5F2A.4.93	121	0	p_T08D10.1.93	86940363	0	p_Y5F2A.4.93	0.58	0	
	p_F19C6.1_BC	121	0	p_R11E3.8_BC	86724370	0	p_F19C6.1_BC	0.58	0	
Head Head Neurons Pharynx	p_C04C3.3_BC	282	0	p_C10G8.7.93	83785321	0	p_C04C3.3_BC	0.86	0	
	p_F30A10.8a_BC	245	0	p_C04C3.3_BC	83779482	0	p_F30A10.8a_BC	0.82	0	
	p_B0252.2_BC	244	0	p_B0252.2_BC	83526292	0	p_B0252.2_BC	0.82	0	
	p_H28O16.1_BC	241	0	p_T04B2.3.93	83267502	0	p_H28O16.1_BC	0.82	0	
	p_Y69A2AR.18_BC	232	0	p_H28O16.1_BC	83148740	0	p_Y69A2AR.18_BC	0.81	0	
	p_T04B2.3.93	225	0	p_F30A10.8a_BC	82811864	0	p_T04B2.3.93	0.8	0	
	p_C10G8.7.93	214	0	p_F42G8.12_BC	82588377	0	p_C10G8.7.93	0.79	0	
	p_C34E10.6_BC	204	0	p_Y69A2AR.18_BC	82482728	0	p_C34E10.6_BC	0.77	0	
	p_Y73F4A.2_BC	192	0	p_C34E10.6_BC	82225210	0	p_Y73F4A.2_BC	0.75	0	
	p_C37E2.1_BC	191	0	p_T27E9.4a_BC	81746410	0	p_C37E2.1_BC	0.75	0	
Spermatheca	p_Y22D7AR.1_BC	76	0	p_Y22D7AR.1_BC	26526289	0	p_Y22D7AR.1_BC	0.41	0	
	p_T23F4.4_BC	65	0	p_T23F4.4_BC	26309369	0	p_T23F4.4_BC	0.36	0	
	p_F09F7.2a_BC	64	0	p_R144.10_BC	26019399	0	p_F09F7.2a_BC	0.36	0	
	p_K11D2.2_BC	64	0	p_F09F7.2a_BC	25990647	0	p_K11D2.2_BC	0.35	0	
	p_Y45F3A.3_BC	64	0	p_Y45F3A.3_BC	25976384	0	p_Y45F3A.3_BC	0.35	0	
	p_B0395.2_BC	61	0	p_K11D2.2_BC	25970313	0	p_B0395.2_BC	0.34	0	
	p_R144.10_BC	60	0	p_B0280.7_BC	25796881	0	p_R144.10_BC	0.34	0	
	p_W02D3.9_BC	59	0	p_T04A8.4_BC	25744341	0	p_W02D3.9_BC	0.33	0	
	p_Y39A1A.15b_BC	58	0	p_B0395.2_BC	25681627	0	p_Y39A1A.15b_BC	0.33	0	
	p_Y50D7A.6_BC	57	0	p_R09F10.6_BC	25662155	0	p_Y50D7A.6_BC	0.32	0	
Hypodermis	p_C10G8.7.93	86	0	p_C10G8.7.93	13805754	5.5e-293	p_C10G8.7.93	0.46	0	
	p_F56H11.1b_BC	60	0	p_Y73F4A.2_BC	13481473	2.33e-265	p_F56H11.1b_BC	0.33	0	
	p_Y73F4A.2_BC	55	0	p_T04B2.3.93	13163656	1.28e-239	p_Y73F4A.2_BC	0.31	0	
	p_T04B2.3.93	51	0	p_B0252.2_BC	13038536	7.56e-230	p_T04B2.3.93	0.29	0	
	p_B0252.2_BC	51	0	p_C04C3.3_BC	12908456	6.6e-220	p_B0252.2_BC	0.29	0	
	p_M6.1a_BC	40	0	p_F56H11.1b_BC	12657838	2.23e-201	p_M6.1a_BC	0.23	0	
	p_C04C3.3_BC	38	0	p_F42G8.12_BC	12639577	4.64e-200	p_C04C3.3_BC	0.22	8.64e-307	
	p_F42G8.12_BC	38	0	p_H28O16.1_BC	12533434	1.74e-192	p_F42G8.12_BC	0.22	1.02e-305	
	p_H28O16.1_BC	32	0	p_T27E9.4a_BC	11862307	5.24e-148	p_H28O16.1_BC	0.19	5.19e-218	
	p_F46F6.2_BC	30	0	p_M6.1a_BC	11857794	9.97e-148	p_F46F6.2_BC	0.18	7.53e-198	
Nerve Ring	p_W08D2.7_BC	78	0	p_F30A10.8a_BC	13811445	1.75e-293	p_W08D2.7_BC	0.42	0	
	p_F30A10.8a_BC	75	0	p_C50D2.7_BC	13780577	8.61e-291	p_F30A10.8a_BC	0.41	0	
	p_C50D2.7_BC	73	0	p_T04A6.1a_BC	13770454	6.33e-290	p_C50D2.7_BC	0.4	0	
	p_F26H0.3.93	72	0	p_Y32H12A.1_BC	13759449	5.80e-289	p_F26H0.3.93	0.39	0	
	p_F12F3.1_BC	71	0	p_R05F9.1a_BC	13748741	4.98e-288	p_F12F3.1_BC	0.39	0	
	p_T04A6.1a_BC	70	0	p_F08C6.7.93	13720353	1.41e-285	p_T04A6.1a_BC	0.39	0	
	p_Y32H12A.1_BC	70	0	p_W08D2.7_BC	13702573	4.79e-284	p_Y32H12A.1_BC	0.39	0	
	p_T01B10.2_BC	70	0	p_T01B10.2_BC	13678716	5.34e-282	p_T01B10.2_BC	0.39	0	
	p_R05F9.1a_BC	70	0	p_C03G5.1_BC	13666025	6.51e-281	p_R05F9.1a_BC	0.38	0	
	p_C27D8.4.93	69	0	p_C34E10.6_BC	13665970	6.58e-281	p_C27D8.4.93	0.38	0	

Table 2S: Top scoring enrichments for different tests, using combined worm sizes. For visualization of head, tail, embryo and vulva see Supplementary Figures 1S - 4S

Statistical Tests on Registered Data, Combined Worms, Top 10 Scores, Cont'											
organ	t-test				wilcoxon			correlation			Overlap
	promoter	t-value	p-value	promoter	ranksum	p-value	promoter	R	p-value		
Vulva	p_C37A2.4_BC	45	0	p_C37A2.4_BC	13417972	4.13e-260	p_C37A2.4_BC	0.26	0		
	p_C04B4.2_BC	35	0	p_F55C7.7a_BC	12863871	1.5e-216	p_C04B4.2_BC	0.21	5.5e-267		
	p_C18G1.2_BC	35	0	p_C18G1.2_BC	12760197	7.58e-209	p_C18G1.2_BC	0.2	7.57e-265		
	p_Y71H2B.10_BC	34	0	p_C37C3.6b_BC	12731398	9.88e-207	p_Y71H2B.10_BC	0.2	1.41e-252		
	p_C27A2.6_BC	34	0	p_C04B4.2_BC	12542540	3.95e-193	p_C27A2.6_BC	0.2	1.05e-245		
	p_F32D1.10_BC	33	0	p_C27A2.6_BC	12536412	1.07e-192	p_F32D1.10_BC	0.19	2.88e-229		
	p_R13A5.6_BC	32	0	p_Y71H2B.10_BC	12421760	1.12e-184	p_R13A5.6_BC	0.19	3.94e-222		
	p_F55C7.7a_BC	32	0	p_C05B5.7_BC	12421533	1.16e-184	p_F55C7.7a_BC	0.19	5.71e-218		
	p_C37C3.6b_BC	32	0	p_F22B3.1_BC	12399955	3.6e-183	p_C37C3.6b_BC	0.18	1.08e-215		
	p_C05B5.7_BC	32	0	p_F32D1.10_BC	12293425	6.67e-176	p_C05B5.7_BC	0.18	1.66e-214		
	p_ZK524.2a_BC	39	0	p_F54C9.4.93	9872625	2e-179	p_ZK524.2a_BC	0.23	0		
	p_Y102A11A.6_BC	39	0	p_W06D4.1_BC	9854145	5.62e-178	p_Y102A11A.6_BC	0.23	2.96e-322		
	p_F19C6.1_BC	38	0	p_F19C6.1_BC	9840381	6.68e-177	p_F19C6.1_BC	0.22	5.9e-306		
p_W06D4.1_BC	35	0	p_K06A5.6_BC	9778463	4.19e-172	p_W06D4.1_BC	0.2	3.06e-259			
p_F54C9.4.93	33	0	p_ZK524.2a_BC	9659728	4.32e-163	p_F54C9.4.93	0.19	1.78e-239			
p_K06A5.6_BC	33	0	p_F41H10.8.93	9652408	1.53e-162	p_K06A5.6_BC	0.19	1.02e-227			
p_F41H10.8.93	30	0	p_C50F7.4_BC	9611734	1.62e-159	p_F41H10.8.93	0.18	2.05e-198			
p_F26A1.14_BC	30	0	p_F54D8.2_BC	9604643	5.42e-159	p_F26A1.14_BC	0.18	1.46e-195			
p_B0218.8.93	29	0	p_F26A1.14_BC	9572478	1.27e-156	p_B0218.8.93	0.17	9.67e-186			
p_B0546.1_BC	29	0	p_F42H10.3.93	9526393	2.92e-153	p_B0546.1_BC	0.17	1.82e-178			
Pharyngeal Intestinal Valve	p_C13G3.3b_BC	72	0	p_C13G3.3b_BC	7028235	2.73e-157	p_C13G3.3b_BC	0.39	0		
	p_JC8.10a_BC	62	0	p_JC8.10a_BC	6975808	1.29e-152	p_JC8.10a_BC	0.35	0		
	p_T23H2.2_BC	60	0	p_K07A9.2_BC	6949718	2.57e-150	p_T23H2.2_BC	0.34	0		
	p_K07A9.2_BC	59	0	p_C51E3.7a_BC	6929560	1.5e-148	p_K07A9.2_BC	0.33	0		
	p_F11D5.3a_BC	58	0	p_F11D5.3a_BC	6920412	9.38e-148	p_F11D5.3a_BC	0.32	0		
	p_F55A12.3_BC	56	0	p_T23H2.2_BC	6914644	2.98e-147	p_F55A12.3_BC	0.32	0		
	p_T04D1.3_BC	55	0	p_F31C3.3_BC	6906153	1.63e-146	p_T04D1.3_BC	0.31	0		
	p_C27H5.8_BC	54	0	p_T04D1.3_BC	6906022	1.67e-146	p_C27H5.8_BC	0.3	0		
	p_F31C3.3_BC	53	0	p_F55A12.3_BC	6901251	4.32e-146	p_F31C3.3_BC	0.3	0		
	p_F54G8.4_BC	52	0	p_C27H5.8_BC	6895982	1.23e-145	p_F54G8.4_BC	0.29	0		

Table 3S: Top scoring enrichments for different tests, using combined worm sizes. For visualization of head, tail, embryo and vulva see Supplementary Figures 1S - 4S

query	head	tail neurons	embryo	vulva
annotations	analdepressormuscle:5 analsphincter:1 bodywallmuscle:6 distaltipcell:1 embryo:5 excretorycell:1 gonadsheathcells:2 headneurons:3 hypodermis:5 intestinal:2 nerving:1 pharynx:6 spermatheca:1 tailneurons:3 unidentifiedbody:1 unidentifiedcells:1 unidentifiedhead:2 unidentifiedtail:1 unknown:3 ventralnerverecord:1 vulvamuscle:2 vulvaother:2	embryo:2 headneurons:1 intestinal:3 unknown:7	analdepressormuscle:1 bodyneurons:1 bodywallmuscle:3 coelomocytes:1 distaltipcell:1 embryo:10 excretorycell:1 headmesodermalcell:1 headneurons:8 hypodermis:2 intestinal:5 intestinalmuscle:1 nerving:4 pharynx:3 seamcells:1 tailneurons:4 unidentifiedbody:4 unidentifiedcells:3 unidentifiedhead:5 unidentifiedtail:3 uterinemuscle:1 ventralnerverecord:3 vulvamuscle:2	analdepressormuscle:1 bodyneurons:1 bodywallmuscle:2 coelomocytes:1 distaltipcell:1 embryo:10 excretorycell:1 headmesodermalcell:1 headneurons:8 hypodermis:1 intestinal:4 intestinalmuscle:1 nerving:4 pharynx:3 seamcells:1 tailneurons:5 unidentifiedbody:3 unidentifiedcells:3 unidentifiedhead:4 unidentifiedtail:2 uterinemuscle:1 ventralnerverecord:4 vulvamuscle:2

Table 4S: Occurrence count of Annotation tags for the top 10 T-Test, shown for different organs

Correlation Coefficient vs. t-test

The way in which we designed our correlation test, will cause this test to give similar results in comparison with the t-test statistics.

The expected value of x will be the mean value of x

$$E(x) = \mu_x \quad (1S)$$

Correlation is defined as

$$\rho_{ij} = \frac{cov(i, j)}{\sigma_i \sigma_j} = \frac{cov(i, j)}{\sqrt{\sigma_i^2 \sigma_j^2}} \quad (2S)$$

The covariance can be rewritten to

$$cov(i, j) = E(ij) - E(i)E(j) = E(ij) - \mu_i \mu_j \quad (3S)$$

When having the value

$$j = \begin{cases} 1, & j \in organ \\ -1, & j \notin organ \end{cases} \quad (4S)$$

and

$$X = \forall i \in organ \quad (5S)$$

and

$$Y = \forall i \notin organ \quad (6S)$$

we can rewrite $E(ij)$ to

$$E(X - Y) = E(X) - E(Y) = \bar{X} - \bar{Y} \quad (7S)$$

Equation 2S can now be rewritten to

$$\frac{\bar{X} - \bar{Y} - \mu_i \mu_j}{\sqrt{\sigma_i^2 \sigma_j^2}} \quad (8S)$$

When comparing this to the equation used to calculate a t-value, we see that

$$\frac{\bar{X} - \bar{Y} - \mu_i \mu_j}{\sqrt{\sigma_i^2 \sigma_j^2}} \approx t = (\bar{X} - \bar{Y}) \sqrt{\frac{n(n-1)}{\sum_{i=1}^n (\hat{X}_i - \hat{Y}_i)^2}} \quad (9S)$$

where

$$\hat{X}_i = (X_i - \bar{X}) \quad (10S)$$

and

$$\hat{Y}_i = (Y_i - \bar{Y}) \quad (11S)$$

The Expected Value of a Wilcoxon rank sum statistic

The total rank sum of a set with size N will be the sum of the range

$$1, 2, \dots, N-1, N \quad (12S)$$

which is equal to

$$\sum_{x=1}^N x = \frac{N * (N + 1)}{2} \quad (13S)$$

The expected rank value of an element x from the ranked set will be its mean (see Eq. 1S)

$$E(x) = \mu_x = \frac{N * (N + 1)}{2N} = \frac{N + 1}{2} \quad (14S)$$

The expected summed rank value of a ranked subset with size S will therefore be

$$S \frac{N + 1}{2} \quad (15S)$$

Master Thesis Proposal

A.1 Master Thesis Proposal

Molecular Imaging allows the detection of specific tissues, diseases or even gene expression using different modalities. Modalities such as CT and MRI contain anatomical information, so that acquisitions of this modality can be registered to a 3D mouse atlas. This registration is important for data normalization and the possibility of follow up studies. Some scanning techniques do not capture any anatomical information. Instead, three (or more) photographic pictures are taken of the animal before it is scanned, in such a way that the photo pictures taken before acquisition are known to be aligned to the x,y,z position of the scan that follows. When it is possible to register a 3D mouse atlas to these three photos, this 3D atlas will automatically also be aligned to the scanned image, which does not have any anatomical information. This creates the following new possibilities:

- Given a registered 3D atlas to a modality such as PET, SPECT, BLT or FMT, anatomical information is incorporated into these modalities, which allows qualitative localization of the measured signal in these modalities.
- For BLT and FMT, it also becomes possible to reconstruct the photon source by solving the inverse problem. Without an anatomical model this would be impossible for BLT and difficult for FMT, because for FMT the forward problem needs to be solved first, before the inverse problem can be solved. For BLT it is impossible to solve the forward problem, because no known input is available.
- Given the aligned 3D atlas, it becomes possible to select a region of interest that can be selected in the three photo images. This ROI in turn will be scanned on the acquisition device which is aligned to the photo pictures.
- Given a registered segmented 3D atlas and a gene expression acquisition, it becomes possible to design statistical tests on temporal and (qualitative) spatial gene expression profiles.

A.2 Thesis Research Proposal

The thesis will comprise two parts. One being an image processing/registration part and the other being the development of statistical tests. Those parts are not necessarily based on the same data source, in fact

it is assumed that for the development and validation of statistical tests, the provided data from MI source will not be available in sufficient amounts. For the development of the statistical tests, we will make use of data acquired from the wormbase database. Data has the form of so-called chronograms (Dupuy 2007). With this data we want to answer the questions:

- given a chronogram, can we make a statement on the qualitative location(s) of the gene with certain reliability?
- given (a) qualitative location tag(s) of a gene, can we make statements on the statistical relevance of the observed signal?

Chronograms will be clustered by using a Pearsons Cross Correlation distance.

For (far) future it could be possible to use gene expression profiles obtained with BLT or FMT with known qualitative gene expression location, to help registering in the 2D/3D registration, by incorporating a 3D photon propagation model in the registration process.

Master Thesis Planning Proposal

Research in steps including estimated time needed.

Research started around March 1st 2008. Projected end date will be December 1st 2008 Planning is based on a workload of 32 hours a week, resulting in 39.5 weeks corresponding to 45 ECT.

Image Registration

Duration	Description
1 month	<ul style="list-style-type: none"> • Make 3D Back-projection out of (at least) two 2D contours. • Rigid registration of a 3D Shape of the same animal (i.e. CT Skin surface) to 3D projection (bounding box) <ul style="list-style-type: none"> – Explore optimization criteria
2 months	<ul style="list-style-type: none"> • Rigid registration of a 3D Atlas (another animal) to the same 3D bounding box.
2 months	<ul style="list-style-type: none"> • Finish with registering non rigid 3D Atlas (In house articulated mouse atlas) by using refinement. <ul style="list-style-type: none"> – Use hierarchical set of joint rotations to obtain rigid skeleton deformations – Generate skin surface from rigid deformed skeleton in 3D atlas. This will result in a non rigid skin surface deformation. – Register calculated skin surface to 3D boundary box by tuning the rotation parameters of the skeleton in an hierarchical manner.

Table B.1: Planning for Image Processing Part.

Note that an optimization function is needed for each step. In each step the optimization function gets more

complex and more parameters need to be optimized. Complexity is reduced by making use of a hierarchical optimization, i.e. one set of parameters is optimized at a time.

Statistical Testing

Duration	Description
1 Month	<ul style="list-style-type: none"> • Make use of data available on WormBase • Find a way to obtain chronograms from WormBase • Translate chronogram images to a structure that can be used for statistical testing, or determine how they can be used as is.
1 Month	<ul style="list-style-type: none"> • Determine if qualitative spatial (organ locations) or temporal (developmental stages) are available.
2 Months	<ul style="list-style-type: none"> • Create a statistical model to determine if genes are significantly differentially expressed in specific organs or in specific developmental stages (age) <ul style="list-style-type: none"> – If qualitative spatial regions are available, segment chronograms in these regions. (horizontally) – If qualitative temporal stages are available, divide chronograms in these regions. (vertically) – Determine if statistical differences are found between these regions. – If statistical differences do no match to qualitative boundaries when available, try to explain. – If qualitative regions are not available, try to find them based on statistical differential gene expression.

Table B.2: Planning for Statistical Testing Part.

1 Month: Write report(s) and prepare final presentation.

APPENDIX C

Working Document on Mouse Paper

Working Document - Martin Wildeman – Last Update January 15, 2009

2D Image / 3D Deformable Mouse Model registration

2D Image / 3D Rigid Mouse Model Registration

To be able to register a 3D deformable mouse model onto 2D images, first a method was developed to register a 3D rigid mouse model. The following problems needed to be tackled to get to a working optimizer.

Problem: Correspondence Problem

We only have two modalities at our disposal to solve the registration problem. First we have a top view and a side view image, next to that, we have a triangle skin surface mesh, based on a CT data set. To get to a registration the skin surface mesh needs to be aligned in 3D space to the 2D mouse contours obtained by segmentation of the mouse pictures.

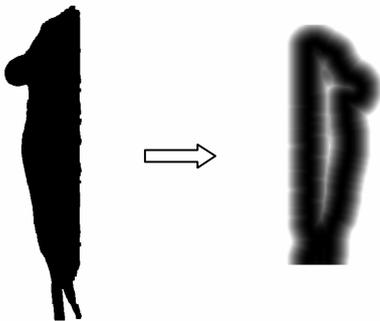
Based on these two modalities it is not possible to determine which vertices in the 3D skin surface model (should) correspond to the points on the contours of the mouse model.

Solution: Distance Matrix

To overcome this problem, instead of trying to find and minimize the distance between corresponding points in the two modalities, a 3D distance matrix is calculated, based on the set of 2D images available. (Currently only 90 degree angle rotation between images is supported)

First, a binary voxel set for each image is created, where the image is projected into the total 3D space. Each slice in the 3D voxel space thereby becomes a copy of the original segmentation in such a way that a stack of copies of one segmentation is created. After that all voxel sets are multiplied with each other, resulting into a 3D bounding box, or 'solution space'

Now that a binary shape estimation of the mouse is calculated, distances to the 0/1 boundary are calculated, using a distance transform function, available in the DIPimage toolbox. Each voxel in the 3D space gets a Euclidian distance value to the nearest border assigned.

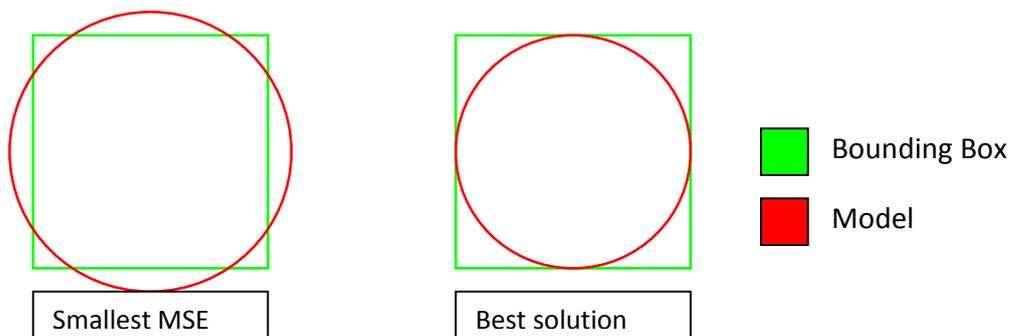


We can now place the 3D model onto the calculated distance matrix and immediately read out the distance per vertex from the boundary of the "optimal" solution. A built-in optimization function, found in the Optimization Toolbox of Matlab is used to find the optimal solution.

Problem: The Square Like Shape of the bounding box vs. The round shape of the mouse.

Because the goal of this research is to align a 3D model to a minimum of 2 side views, the solution space in the form of projected segmentations (a.k.a. the bounding box) will give a square like shape as a result. The more viewpoints are obtained, the less square the bounding box will be. Because of the assumption though the light beams onto the mouse and from the mouse during acquisition of the photo are parallel, view from top-bottom and left-right are assumed to have the same information and thus count as one.

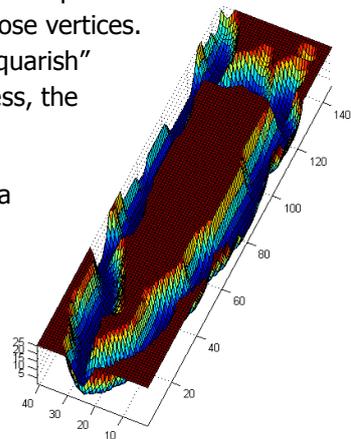
As a result an optimal fit, will give the lowest mean squared error, which is not necessarily the best solution. For example see schematic below.



Solution: Introduction of angles and "trenches"

If only vertices are taken into account that point to the same direction in space in both the model as the bounding box, the fit will become better for those vertices. The total error over all vertices will become worse, because of the "squarish" nature of the bounding box. i.e. The further in the optimization process, the more weight the 'contour-vertices' get.

A similar effect can be seen when the distance transform is cut of at a certain value. When all values above 5 voxels away from the 0 distance surface get the value 5, the Jacobian that is calculated at those locations will be zero causing the optimization algorithm to stop at those points. Especially in the corners of the bounding box, the distance can be larger than this threshold and thus will not be taken into account for the optimization.



Problem: Because of the discrete nature of the distance matrix, it is not possible to use the built in Jacobian estimator of the lsqnonlin function (Least Squares Non Linear Model fitting) This estimator makes such small discrete increments, that there is no difference in fitness and thus no improvement estimated.

Solution: The Jacobian Matrix is estimated manually, by taking large(r) steps. To get a good estimation, to calculation for each parameters are made, $-\delta x$ and a $+\delta x$

Currently the following stepsizes are used:

Used Translation step size	0.5
Used Rotation step size	5
Used Scaling step size	0.01

These values are taken manually. A more intelligent way of determining the step sizes should be thought of, but current settings seem to work well.

Problem: Matlab is very slow when using for loops, but very fast when using matrix operations. To compare a list of x,y,z values with the 3 dimension distance matrix (voxel field), an iteration through all x,y,z values is needed.

Solution: By addressing the voxel space differently a huge speed up is obtained. A voxel that would 'normally' be addressed with `matrix(10,23,12)` can also be addressed as, given that we have a matrix of $20 \times 40 \times 50$ fields big, `matrix(10+(23-1)*20+(12-1)*20*40)` of `matrix(9250)`. The last value can be calculated first, such that a list of alternative addresses is obtained. With this alternative list, a direct 'voxel readout' can be achieved. The for loop is completely removed and transformed into a matrix address vector. This vector can be used to do a single readout on the matrix and assignment to an error vector.

Problem: Digimouse is a rigid model. We need a semi non rigid registration.

Solution: To be able to deform the mouse skin surface non rigidly but to also reduce the degrees of freedom as much as possible, we will make use of the rigid nature of the mouse skeleton. Every joint has 1 to 3 degree of freedom in rotation, every bone has 3 degrees of freedom in scaling and location/translation is restricted by connectivity in the joints.

The program Amira was used to manually segment the skeleton into separate bones, where the spine and pelvis were left in one piece or bone. Organs can be roughly interpolated between ribs, pelvis and spine.

The rotation or pivot points in the joints were manually estimated, as well as the local orientation.

Axis are defined as follows: axis has it's origin in the pivot points. The longest part of the bone will be the x-axis. Y-axis will be orthogonal on that, preferably pointing to the direction of the fixed bone. When that is not possible, it will point up as much as possible. (So there is no rotation around the x axis.)

Table 1: Manually Selected Points

	Origin			A Point on X			A Point on Y		
Spine	188,5	132,5	456,5	188,5	132,5	702,5	188,5	202,5	702,5
Skull	200,5	130,5	722,5	204,5	68,5	928,5	208,5	172,5	740,5
Left_Upper_Arm	246,5	122,5	706,5	322,5	60,5	704,5	252,5	132,5	706,5
Left_Lower_Arm	322,5	60,5	704,5	316,5	52,5	818,5	246,5	122,5	706,5
Left_Hand	316,5	52,5	818,5	316,5	52,5	848,5	326,5	62,5	818,5
Right_Upper_Arm	138,5	118,5	712,5	72,5	58,5	674,5	120,5	138,5	710,5
Right_Lower_Arm	72,5	58,5	674,5	88,5	58,5	780,5	138,5	118,5	712,5
Right_Hand	88,5	58,5	780,5	88,5	58,5	816,5	88,5	70,5	780,5
Right_Upper_Leg	240,5	136,5	234,5	336,5	64,5	254,5	264,5	150,5	232,5
Right_Lower_Leg	336,5	64,5	254,5	286,5	58,5	104,5	240,5	136,5	234,5
Right_Foot	286,5	58,5	104,5	388,5	58,5	104,5	336,5	64,5	254,5
Left_Upper_Leg	140,5	120,5	232,5	52,5	48,5	234,5	122,5	140,5	228,5
Left_Lower_Leg	52,5	48,5	234,5	100,5	40,5	94,5	140,5	120,5	232,5
Left_Foot	100,5	40,5	94,5	18,5	26,5	128,5	52,5	48,5	234,5

Table 2: Corrected Values

	Origin			A Point on X			A Point on Y		
Spine	188,50	132,50	456,50	188,50	132,50	702,50	188,50	202,50	456,48
Skull	200,50	130,50	722,50	204,50	68,50	928,50	208,40	174,03	735,43
Left_Upper_Arm	246,50	122,50	706,50	322,50	60,50	704,50	253,80	131,44	706,47
Left_Lower_Arm	322,50	60,50	704,50	316,50	52,50	818,50	246,59	122,62	704,86
Left_Hand	316,50	52,50	818,50	316,50	52,50	848,50	326,50	62,50	818,50
Right_Upper_Arm	138,50	118,50	712,50	72,50	58,50	674,50	120,96	138,91	710,76
Right_Lower_Arm	72,50	58,50	674,50	88,50	58,50	780,50	131,42	118,50	665,61
Right_Hand	88,50	58,50	780,50	88,50	58,50	816,50	88,50	70,50	780,50
Right_Upper_Leg	240,50	136,50	234,50	336,50	64,50	254,50	256,35	156,61	230,80
Right_Lower_Leg	336,50	64,50	254,50	286,50	58,50	104,50	255,22	138,27	278,65
Right_Foot	286,50	58,50	104,50	388,50	58,50	104,50	286,50	64,50	254,50
Left_Upper_Leg	140,50	120,50	232,50	52,50	48,50	234,50	123,43	141,26	228,48
Left_Lower_Leg	52,50	48,50	234,50	100,50	40,50	94,50	131,91	121,93	257,55
Left_Foot	100,50	40,50	94,50	18,50	26,50	128,50	139,67	63,38	198,36

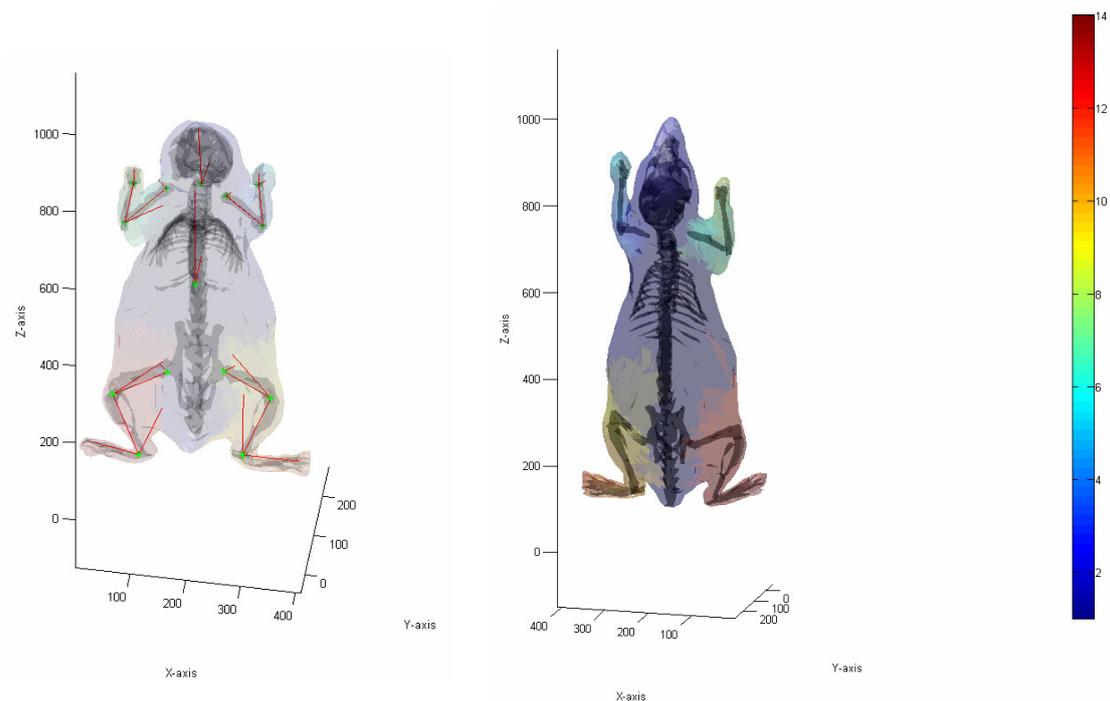
Problem: The skin surface should be estimated based on skeleton.

Solution: Estimate which vertices belong to which bone. Do this by making a nearest neighbor calculation. Transformation applied to bone of closest vertex, will also applied to this vertex of the skin. Eventually also a weighted transformation can be applied, based on distance from pivot point and orientation to pivot point. (Papadimitris et al.)

If skin surface is not sub sampled, right now $9.7 \cdot 10^{10}$ distances need to be calculated. This needs to be reduced. Skeleton probably can be simplified.

First vertices appointed to skeleton will get same transformation as corresponding bone in the skeleton. At the joints, some correction may be needed, but currently the optimization will use 'unsmoothed' skin surface for optimization.

In stead of subsampling, we calculated the distances in parts. After all skin vertices we label, we discarded all skeleton vertices, because these are not used anyway for the registration.



Problem: Some vertices of the skin are appointed to the wrong skeleton bone.

Solution: Euclidian distance appears not to be strong enough. (see Figure 1) Do a manual segmentation, or try to remap all labels that have less than 2, (or 1) neighbors with the same label.

Even better solution: I changed a \geq constrain into a $>$ constrain. All artifacts are gone. (Figure 2)

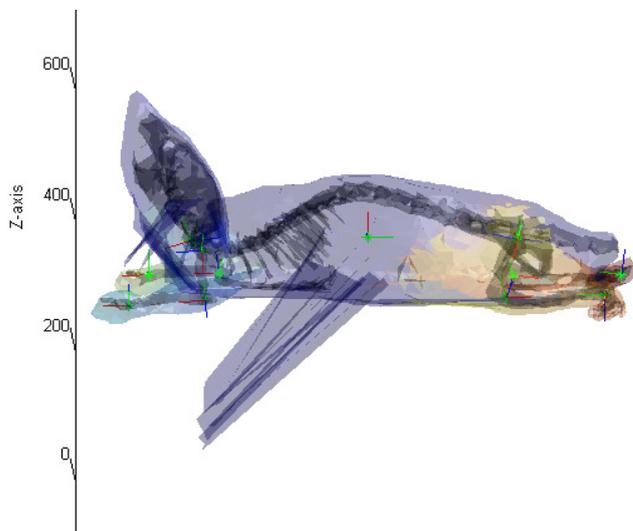


Figure 1: When tilting the head, some point near the abdominal are also rotated. This suggests that they are falsely assigned to the skull. This could be due to noise of the skull.

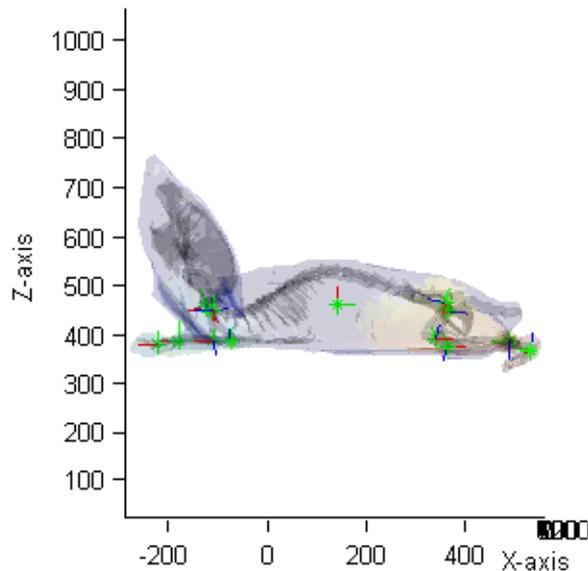


Figure 2: All artifacts are gone

Problem : Rotations should be defined unambiguously, to have a clear meaning.

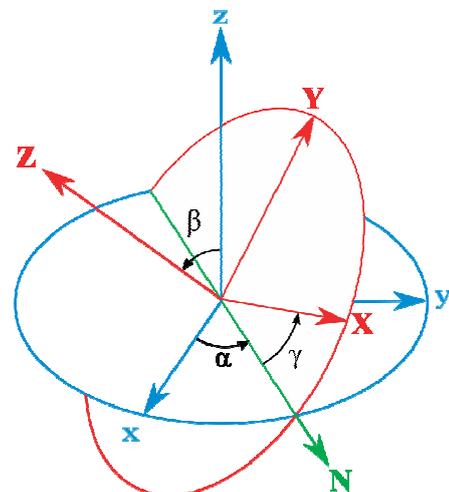
When rotating sequentially around the x and y axis, it is possible to end up with a rotation around the z axis, without defining this angle.

Solution: Use Euler Angles to define local axis. To reduce degrees in freedom of some rotations, set some Euler angles fixed to 0. With the local axis as reference.

Problem: Euler Angles are complicated when using a hierarchical transformation model. For scaling in a certain direction they are well suited, but for local rotations Euler Angles do not suffice.

Update: In view of consistency, the Euler Angles have been converted to quaternions also. Angle estimation is done by using a linear search algorithm.

Solution: For local rotation, i.e. rotations in the joints, rotations around the locally defined axis, in the form of unit vectors is needed. Quaternions are perfectly suited for this problem. With Quaternions, rotation Matrices can be constructed that will result in a rotation around a unit vector.



For both scaling and rotation, points need to be mapped to the global origin. So when manipulating the full mouse model, all the pivot points need to be updated, to be able to translate single rigid bodies to the origin. The pivot points from table 2 are used in the process.

The mouse manipulation algorithm is described in the topic "Mouse Manipulation Algorithm Used"

Problem: The new algorithm MouseManipulator works very well and is fairly efficient. Unspite a fairly fast computation (less than a second), the calculation of the Jacobians becomes very costly though. For each parameter to optimize, each iteration a Jacobian has to be calculated. For a Rotation/Scaling/Translation operation on all three axes, this mean 18*less than a second. For 3 iteration already 30 seconds are needed.

Solution:

While the skin transformation is a non rigid operation, it is in fact a combination of rigid transformations. It must be possible to do small updates to the model, because all updates are linear combinations of transformation Matrices.

Also it should be noted that when the local axis are updated after each operation the following statement holds:

+90 deg x → - 90 deg x → + 90 deg y is equal to **+ 90 deg x → + 90 deg y → - 90 deg x**

Normally, when the rotation axes are fixed, the second equation would yield a rotation around the x, y AND z axis, because after the first two operations, the x axes would be located on the position where originally the x axis was located. When updating the local axis with each operation, this is not the case, and a rotation around the x axis will always be a rotation around the same axis with respect to the rigid body. The first equation would yield a rotation around the y axis, because the first two operations in both cases cancel each other out.

Subproblem:

When applying incremental updates to the mouse model, the Euler Angles calculated before need to be updated as well, to be able to do incremental scaling operations with respect to the local axis. Some way has to be found to update these euler angles, or some other way of scaling along the axis has to be found.

Workaround:

In order to save calculation time and in order to keep the mouse model consistent, I have decided to undo rotations first, by applying matrix multiplications in reverse order. After that the new rotation angles are calculated.

Problem:

The main part of calculation time is consumed by the calculation of sines and cosines.

Possible solutions:

I've contacted Jan Bot in order to compile Matlab code with a C compiler. This should be possible within a week.

Bart Witteman came with the suggestion to convert the sine functions to a lookup table, in order to improve speed.

Both approaches didn't work out and we agreed on the fact that performance issues were not of importance. We therefore focused on accuracy and precision instead of speed for the algorithm.

Problem:

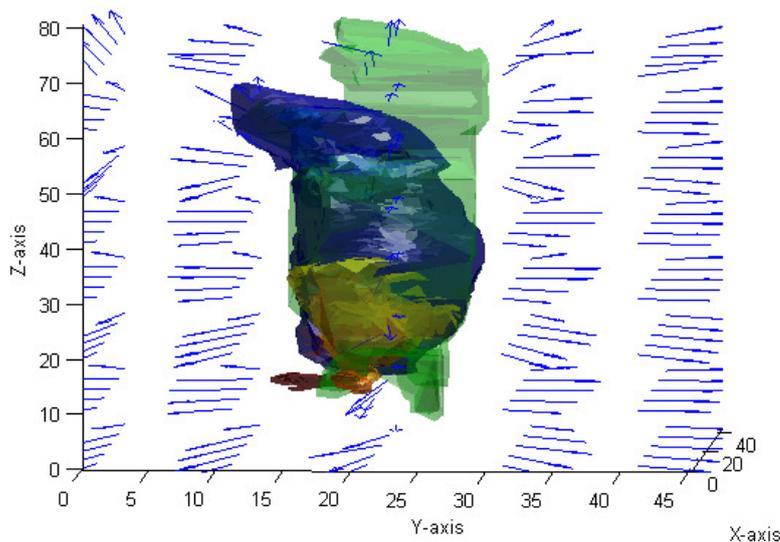
The head of the mouse is squeezed into the area where the paws should go. This problem cannot be solved by making use of angles or 'ditches', because both yield a suboptimum.

Solution:

Try to come up with an extra penalty (which is also cheap in terms of computational power). For instance, minimize the difference between maximum x,y,z and minimum x,y,z . This will force the mouse model into an as large as possible area, during the first optimization step, i.e. the global scaling.

Update:

Instead of introducing an extra constraint, the initialization is set outside of the bounding box. This way optimization is done by "shrinking" the mouse to an optimal fit. Also some brute force large rotation are tried, before starting optimization step, to start with the lowest error and thereby avoiding local optima.



Problem: A reasonable set of rotation restrictions need to be set. I.e. try to estimate feasible rotation angles.

Solution: Set these parameters manually. A good estimation will hopefully suffice.

Current set of restrictions:

Table 3: Rotation and Scaling restrictions.

	Min Scaling	Max Scaling
All	0,8 *	1,2 *

	initial	initial				
	Min Rot x	Min Rot y	Min Rot y	Max Rot x	Max Rot y	Max Rot z
Spine	-20	150	-30	20	210	30
Head	-5	-10	-20	5	10	20
Left Upper Arm	-10	-10	-40	30	30	60
Left Lower Arm	0	0	-20	0	0	90
Left Hand	-5	-5	-20	5	5	20
Right Upper Arm	-10	-10	-40	30	30	60
Right Lower Arm	0	0	-5	0	0	90
Right Hand	-5	-5	-20	5	5	20
Left Upper Leg	-10	-30	-40	10	60	10
Left Lower Leg	0	0	-30	0	0	70
Left Foot	-5	-5	-25	5	5	110
Right Upper Leg	-10	-30	-40	10	60	10
Right Lower Leg	0	0	-30	0	0	70
Right Hand	-5	-5	0	5	5	125

Problem: When updating the mouse model, a lot of local suboptima exist, in which the optimization algorithm can get stuck. It is not feasible to apply a full search algorithm.

Solution: Instead of doing a full search optimization, try to get a good fit as initialization, by applying an 'intelligent' brute force initialization. Scan through the possible feasible solution space to obtain this initial point. Note that this initialization is a very expensive one and the for a refinement of the search space where n is the refinement factor, the increase in needed computation power will be n^7 .

Problem: The results need to be compared with some ground truth

Solution: Take a CT surface as a ground truth. Generate synthetic 2D image segmentations to construct a 3D bounding box with. Use this bounding box as the optimization function and calculate the real error by comparing the optimized Digimouse skin surface with the initial CT skin surface.

	Skin 1	Skin 2	Skin 3
Mean Absolute error	5.933466e+000	4.223713e+000	5.925363e+000
Mean Error	1.272084e+000	2.821863e+000	-1.336364e+000
MSE	5.726520e+001	2.827605e+001	7.077466e+001

Figure 3: Errors obtained after optimization. Errors are in Voxels

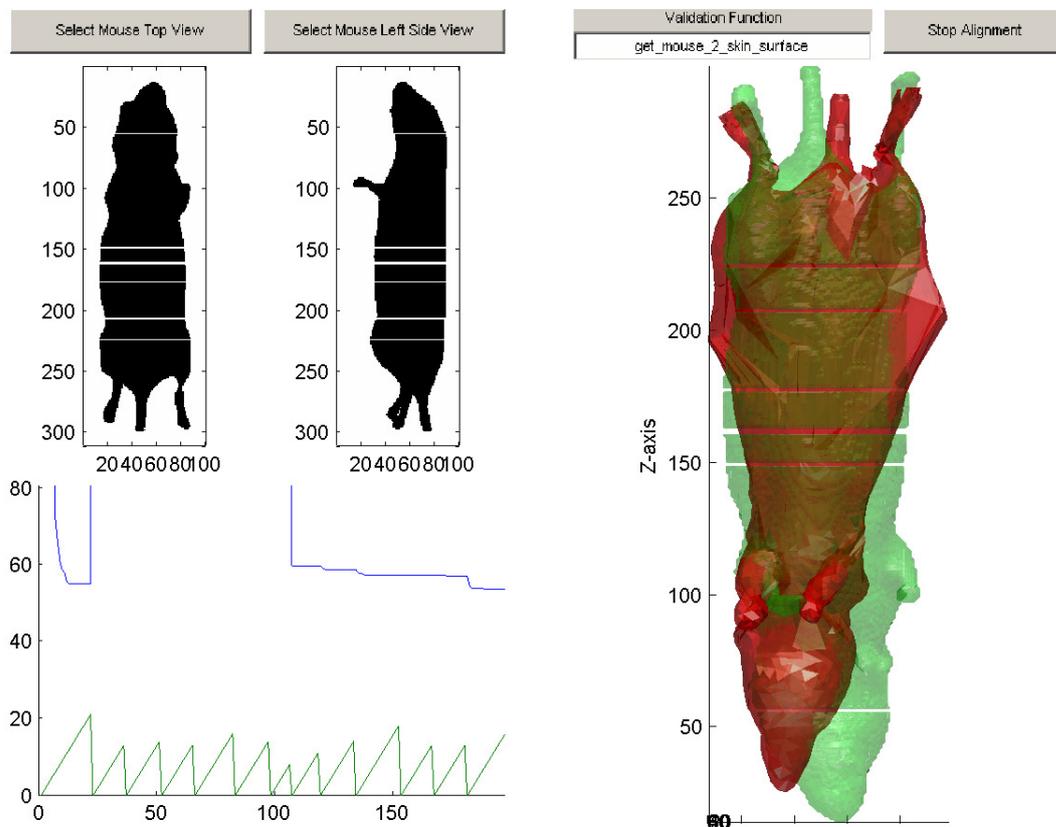


Figure 4: A final 'fit' is found which is close to the true location around the feet, but around the head, the error is larger.

Problem: The error and thus the optimum, is calculated, by looking at the vertices lying on the surface. But the error we're interested in is the overlap in volume.

Solution: It is not feasible to take the voxelspace as an argument for the fitness space. Instead it might be possible to use a point distribution on the surface which is not uniform, but related to to "volume" covered by the vertices on the skin. This will not be applied in the method, but is merely a recommendation.

Note: It seems that an rotation delta of 1 in the calculation of the Jacobian is not course enough. When using a delta +/- of 5 the optimization will get stuck in an optimum less fast. Note also that the smallest rotation still giving a difference where a difference is expected is the best delta increment possible, but in the current setup it is not possible to use a dynamic step size. Since we have a discrete solution space, the only way to obtain a smaller step size is to make an interpolation of each vertex. This is very computational expensive, because it will need a lookup of 8 values for each vertex, resulting in a final weighted value.

Update:

Instead of using a dynamic change in step size, we decided to use a discrete change. After an optimum has been reached, we decrease the rotation step size and start again. This gives a small improvement, which could especially be useful for the Digimouse optimization, due to its hierarchical nature.

Problem: The body of the mouse is not correctly registered because the limbs give a larger error, when registering.

Solution: Only take into account parts of the body that are optimized when calculating the error. The labeling of vertices is already present. Implementation in matlab is by making use of the function 'find'. All vertices that do not have the correct label will be set to value 0. Since the labeling is a constant, setting these vertices to some constant value, will not have an effect on the derivatives and thus on the optimization process. Setting them to zero though, will give a better indication of skin surface error, during optimization.

Problem: (Euler) Angles are a dependent parameter set. If an adjustment is made in one angle, chances are that the other parameters change a lot.

Solution: Instead of using Euler angles, Use x-y-z rotations. This will not prevent the problem, but since small adjustments are expected, also smaller deviations in the rotations are expected. This would probably give rise to less problems, since we also make the assumption that the initial orientation of the 3D surface is roughly right.

Problem: It is not possible to obtain rotation parameters from a transformation matrix, since the solutions are not unique. Rotation parameters are dependent set. Another measure for rotation error is therefore needed.

Solution: Instead of directly comparing the angles applied for the transformation, calculate the angle between a vector $[1 \ 1 \ 1]^T$ * transformation Matrix. The smaller this angle is, the better the registration apparently is.

Problem: The initialization of the left leg goes wrong in many cases.

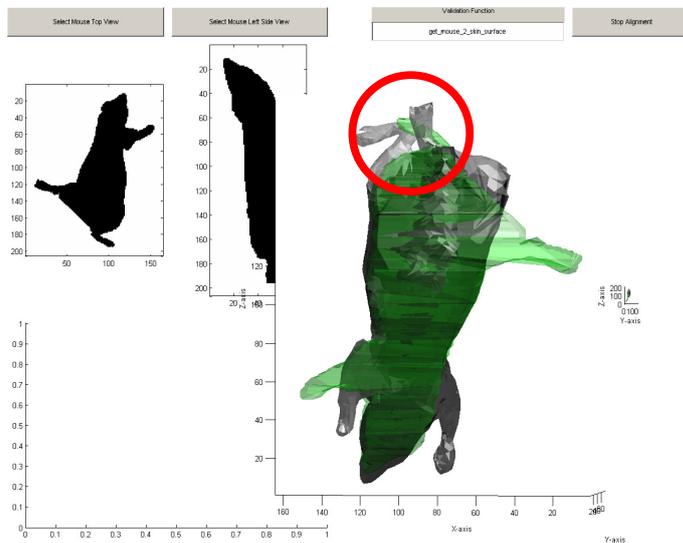


Figure 5: Local optimum, The Left leg has a degree of freedom that allows for initialization in the position of the right leg.

Solution: Make the search space smaller and find a way to make sure the mouse does not end up in infeasible positions.

Extra Advantage: If the search space becomes smaller. The time to search becomes less as well.

While we have changed the feasible rotation space, we still observe that the registration of the left leg systematically more often is registered incorrectly. After searching for bugs in the code for a long time and after many testing, we concluded that the error is not in the code. When trying to make some nice visualizations in Amira, we accidentally bumped into a labeling error, that might explain the registration errors of the left foot. Because of the wrong labeling in the ankle joint, the closest point labeling that we apply will assign some vertices at the bottom of the left foot, to the lower leg bone, resulting in unpredictable directions are the vertex normals, when applying rotations to the ankle joint. The labeling error is visualized in Figure 6.

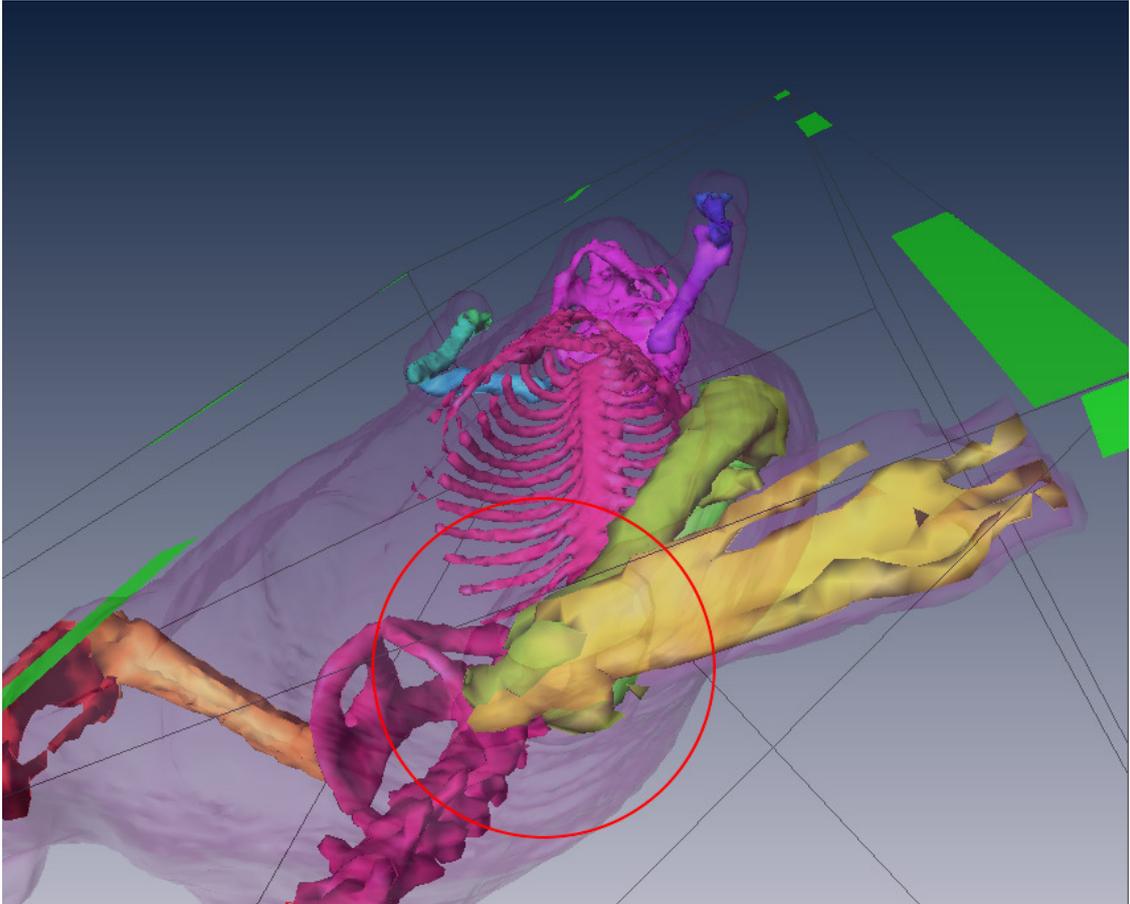


Figure 6: Wrong Labeling of the ankle joint.

Varying rotation step size in estimation of Jacobian.

After running the optimization with the new parameter set (Table 4) the results looked more promising, but still not good. Inspection of the results show the right arm was wrongly estimated many times. A bug was found in the code where in the hierarchical optimization step of the right arm, the rotation restrictions of the left arm were used. This now has been corrected.

Table 4: Final Rotation and Scaling Restrictions

	Min Scaling	Max Scaling				
All	0,8 * initial	1,2 * initial				
	Min Rot x	Min Rot y	Min Rot z	Max Rot x	Max Rot y	Max Rot z
Spine	-10	170	-10	10	190	10
Head	-10	-10	-20	10	10	20
Left Upper Arm	-10	-10	-40	10	30	60
Left Lower Arm	0	0	-20	0	0	90
Left Hand	-5	-5	-20	5	5	20
Right Upper Arm	-10	-10	-40	30	30	60
Right Lower Arm	0	0	-5	0	0	90
Right Hand	-5	-5	-20	5	5	20
Left Upper Leg	0	0	-20	0	20	20
Left Lower Leg	0	0	-20	0	0	40
Left Foot	-5	-5	-25	5	5	60
Right Upper Leg	0	0	-20	0	20	20
Right Lower Leg	0	0	-20	0	0	40
Right Hand	-5	-5	0	5	5	60

Problem: Is normalization of the thresholded distance map useful?

Answer: Not really. The energy function does not really change from optimization.

$$\min \frac{1}{n} \sum_{v=1}^n \left(\frac{DM(\text{round}(T(v, \Theta)))}{D_{\max}} \right)^2 = \min \frac{1}{n} \sum_{v=1}^n \frac{(DM(\text{round}(T(v, \Theta))))^2}{D_{\max}^2}$$

$$= \frac{1}{D_{\max}^2} \min \frac{1}{n} \sum_{v=1}^n (DM(\text{round}(T(v, \Theta))))^2$$

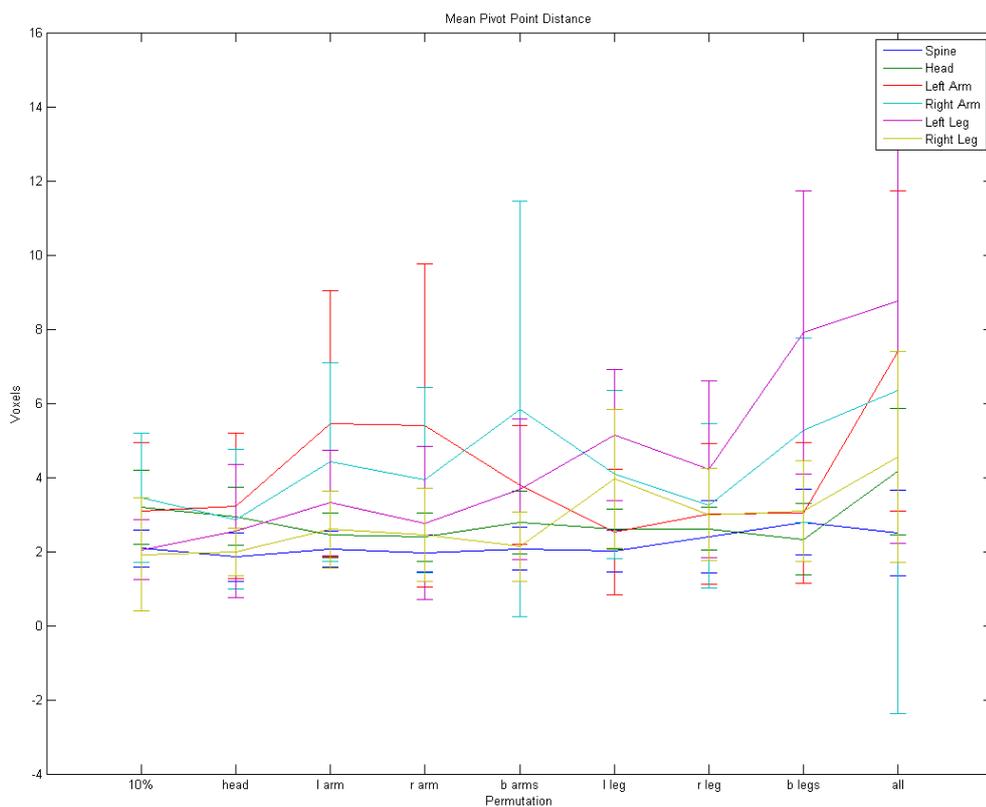
The only difference that could affect the optimization is that the penalty function that is given, is the same as D_{\max} . When this penalty is not normalized to 1, the optimization could be affected by the preset precision (the error stop criterion). When a vertex is given a larger penalty, the precision of the algorithm can be smaller (larger error stop criterion).

On the other hand, when the distance part of the energy function is normalized, it loses its direct meaning, being distance. When D_{\max} is 25 for example, a distance of 1 is normalized to

1/625. When D_{max} is set to 5, a distance of 1 will be normalized to 1/25. This would mean that the optimization algorithm would need more precision when D_{max} is set to a higher value.

The penalty of the angle needs to be set to the maximum possible distance, D_{max} , because otherwise, the event could occur that vertices having a distance of D_{max} would benefit from an angle penalty, which is set to a smaller error than D_{max} .

To summarize: When D_{max} is relatively larger, the optimization algorithm would need more precision for the skin error part when normalized and relatively less precision for the angle penalty when not normalized. The inverse holds for a relative smaller D_{max} . Since for fine tuning the allowed angle is set to a small value, the angle penalty part counts for a large part in the energy function. It can therefore be argued that normalization is slightly favorable.



Problem: Even with the incorporated angle penalty, the scaling factor of the registration is overestimated. Especially when using the hierarchical model, it is of importance to register the main body as good as possible. Otherwise the positioning of the limbs will not finish successfully.

Solution: We incorporate extra knowledge. Since we know that the bounding box is generated by the surface of the mouse, we also know that all vertices have to lie within that bounding box. We therefore can penalize vertices that are on the outside of the bounding box. The right parameter setting needs to be found though and it could be conflicting with the angle penalty. All vertices that lie outside of the bounding box are multiplied with this penalty. The effect of alpha is 5 is shown in Figure 7 and Figure 8.

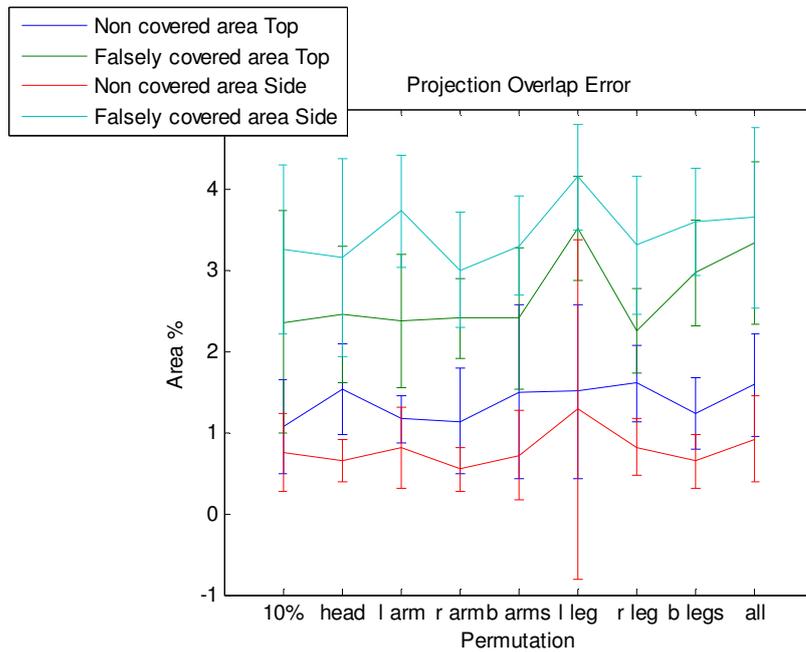


Figure 7: The composition of the Dice index, alpha is set to 5

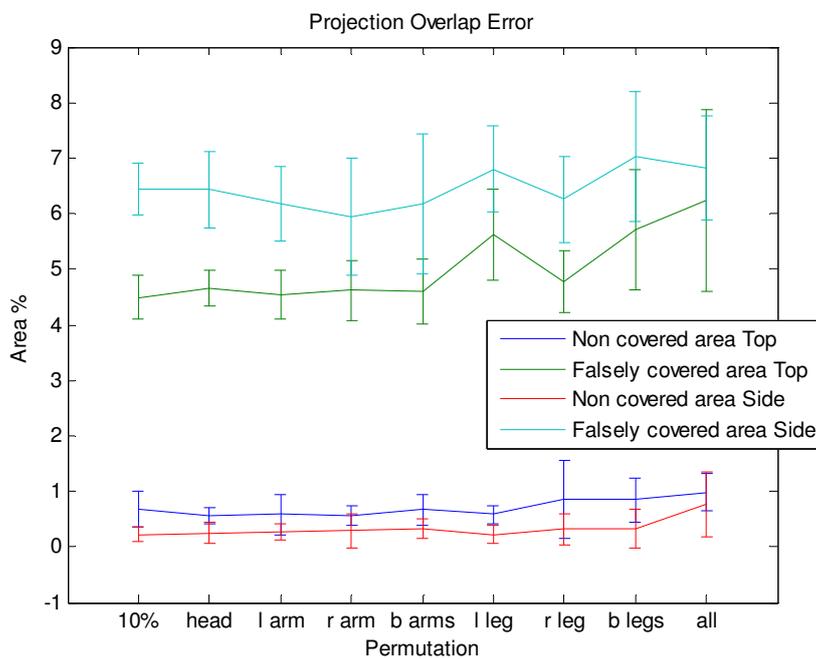
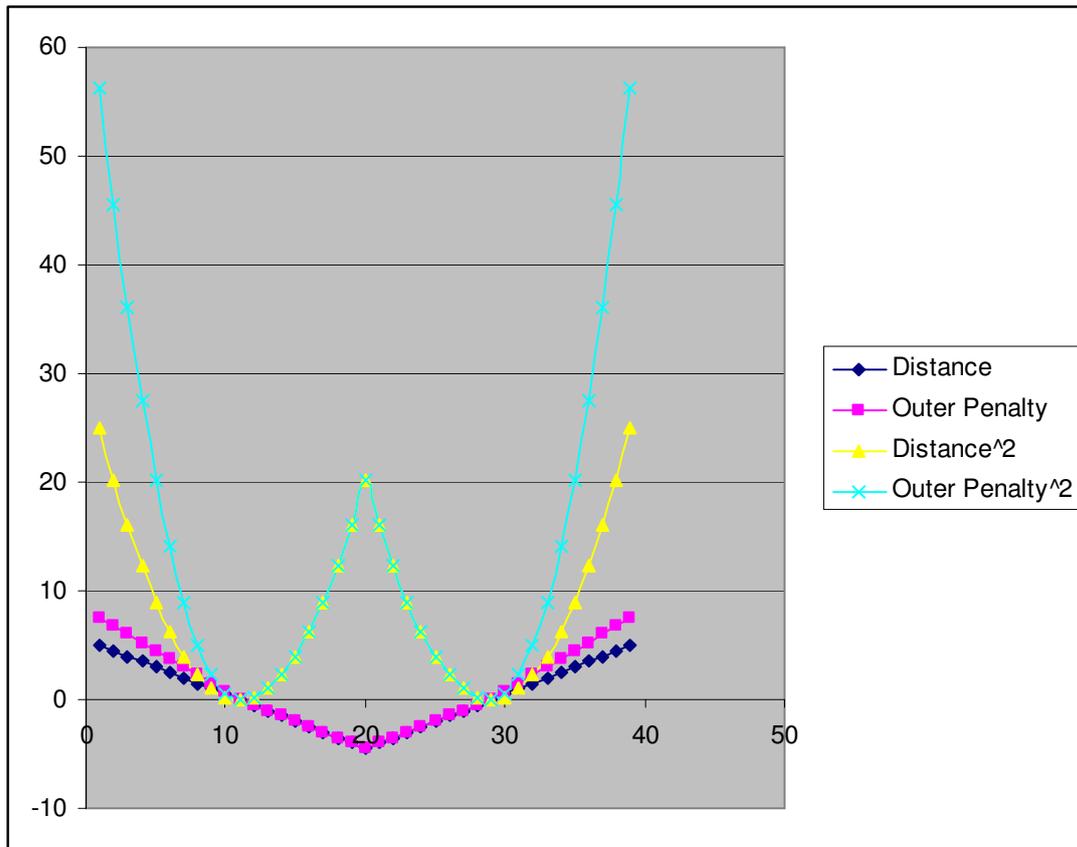
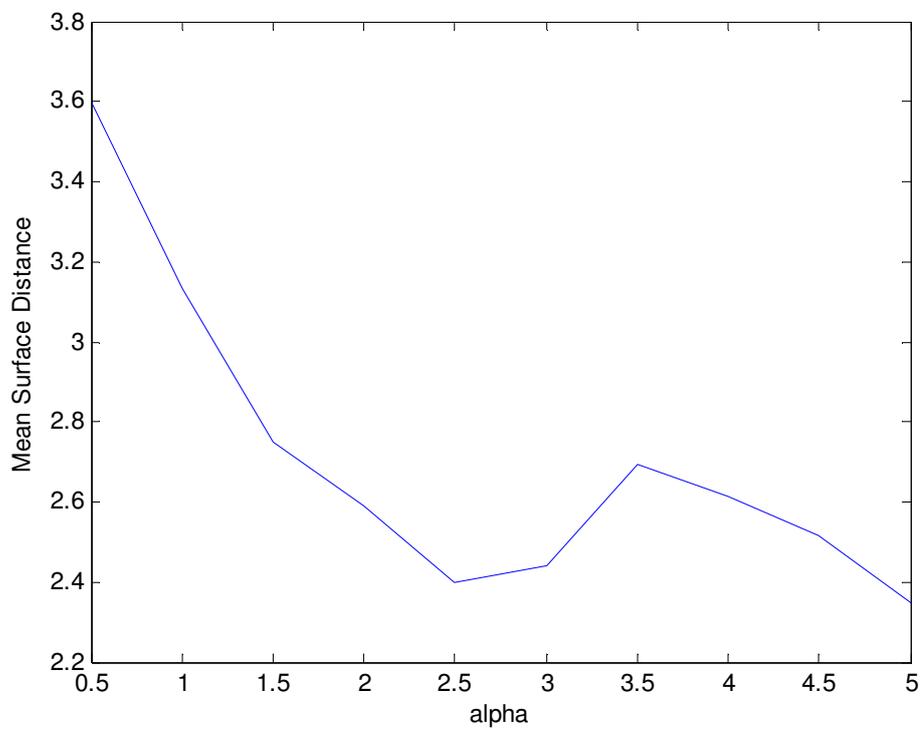
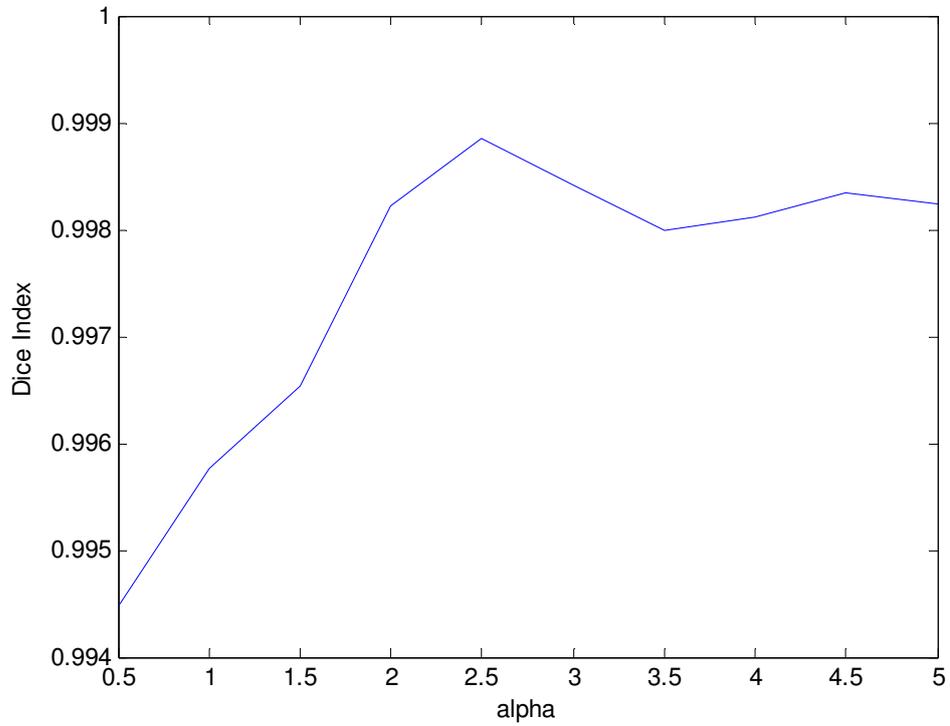


Figure 8: The composition of the Dice index, alpha is set to 1



We tuned this parameter in such a way that we obtained the best Dice index for a real data example view two side views. The parameter

Preliminary test of best value for α . For this one mouse an alpha of 2.5 seems optimal. Oddly enough overall results also show to be the best we could achieve with alpha is 2.5

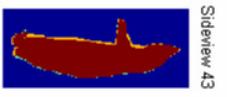




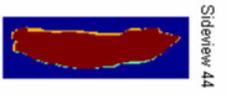
Sideview 41



Sideview 42



Sideview 43



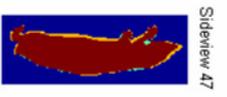
Sideview 44



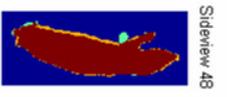
Sideview 45



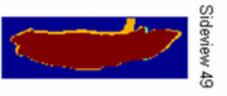
Sideview 46



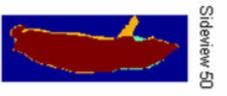
Sideview 47



Sideview 48



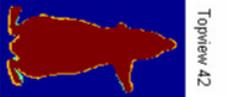
Sideview 49



Sideview 50



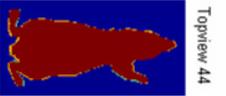
Topview 41



Topview 42



Topview 43



Topview 44



Topview 45



Topview 46



Topview 47



Topview 48



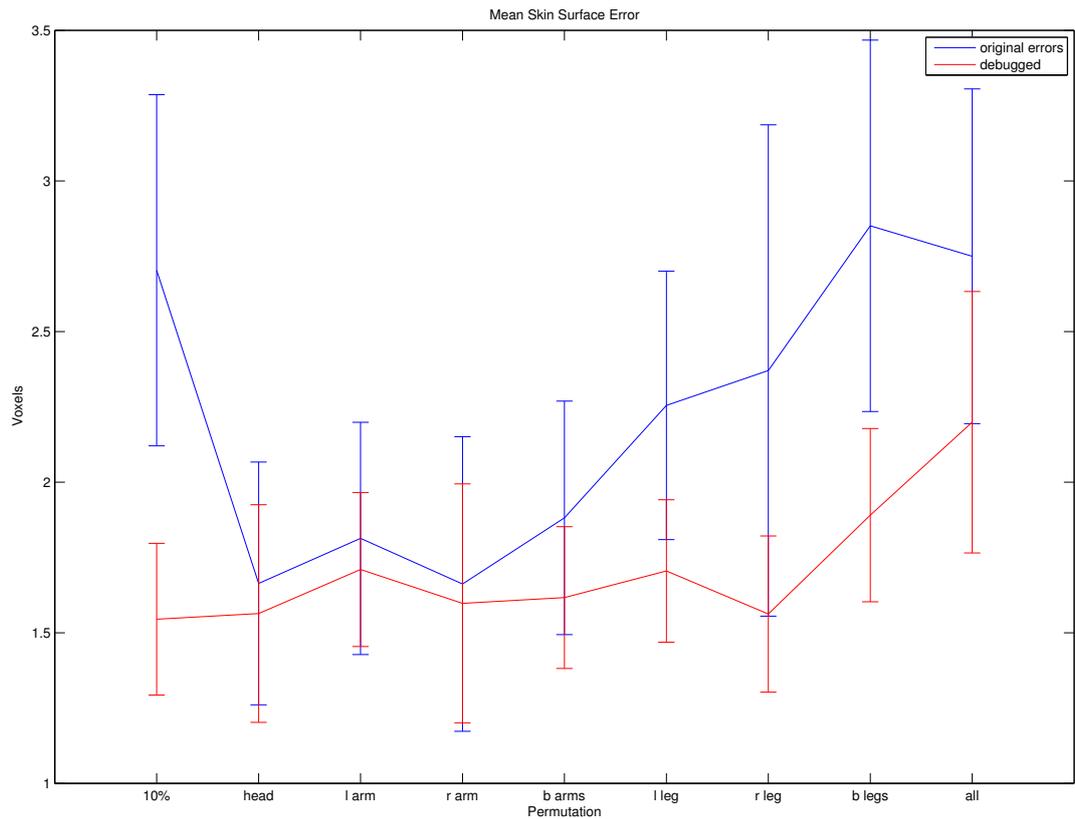
Topview 49



Topview 50

After 'fixing' the following two bugs, the errors went down a bit.

- Freedom of rotations was unrealistically large. Therefore virtually each initialization would start in the proximity of the spine skin surface.
- For initializing and optimizing the right arm, the restrictions of the left arm were used. This gave a discrepancy between the random generation of the mouse model and the registration.



The golden standard doesn't seem to be gold at all. We can thus not quantify the performance of our algorithm. Instead of that, we compare the Dice indices with those of the Gold standard to make it plausible that our algorithm at least gets a better registration than what is obtained with the golden standard registration.

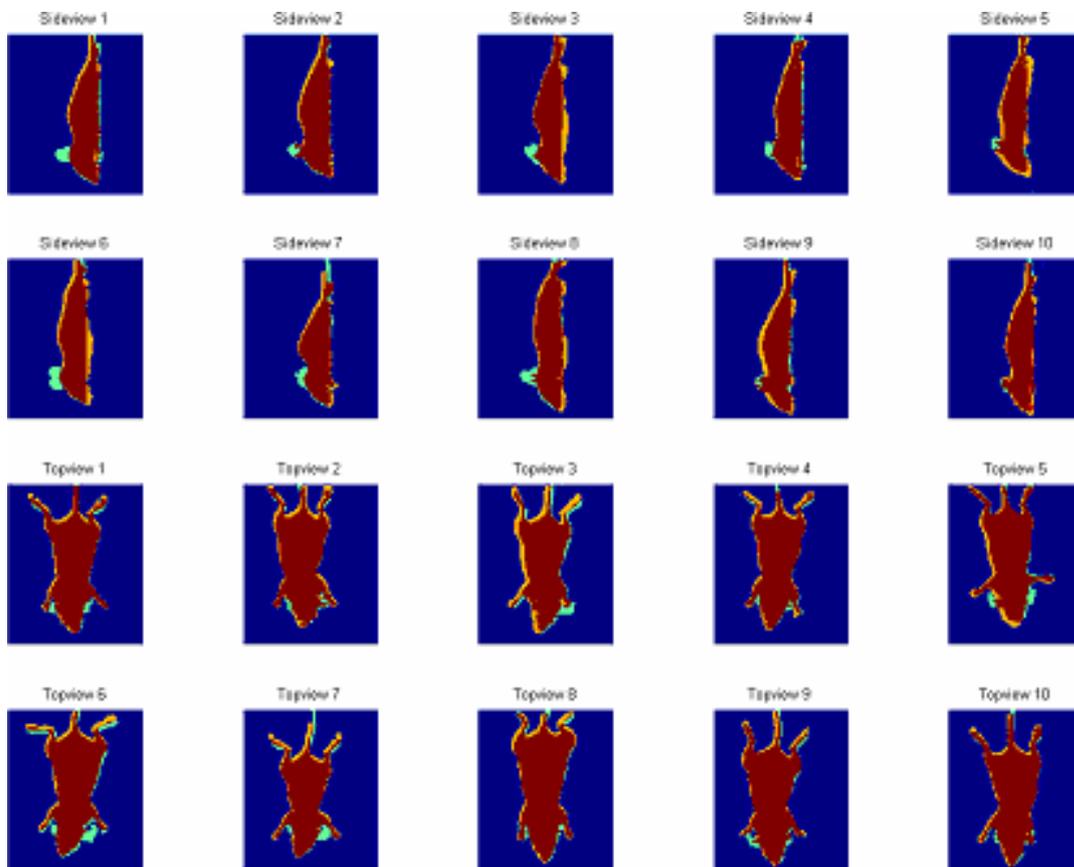


Figure 9: Golden Standard

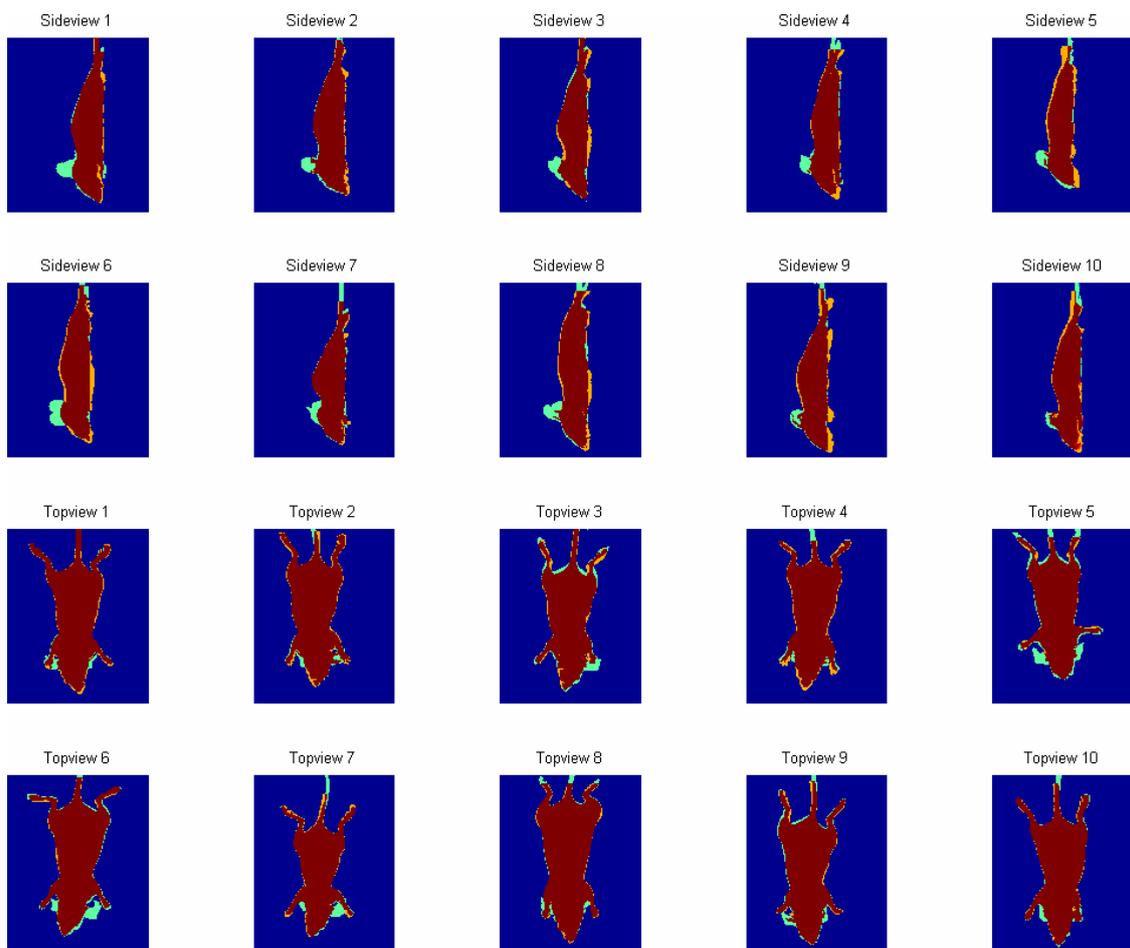
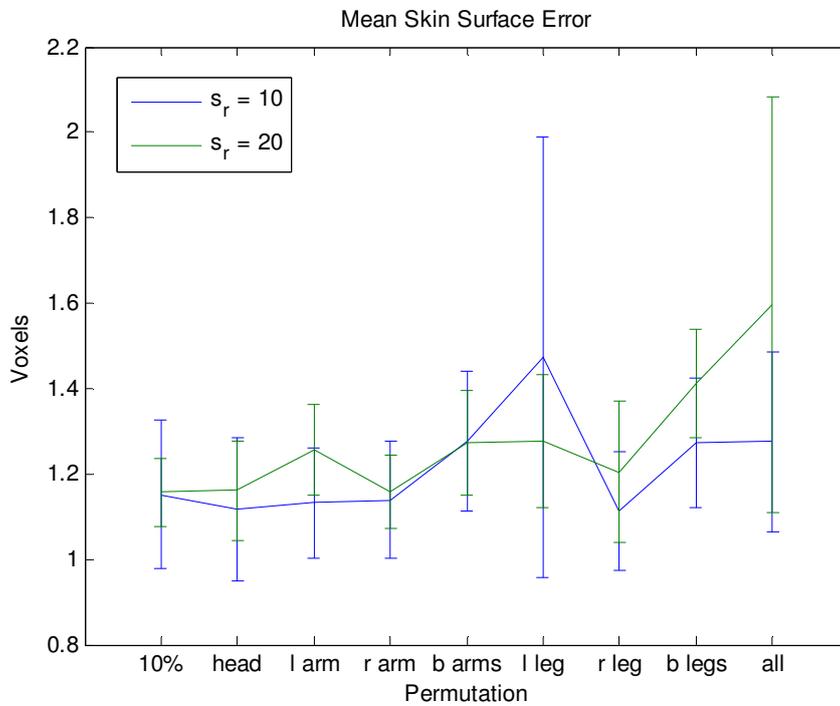
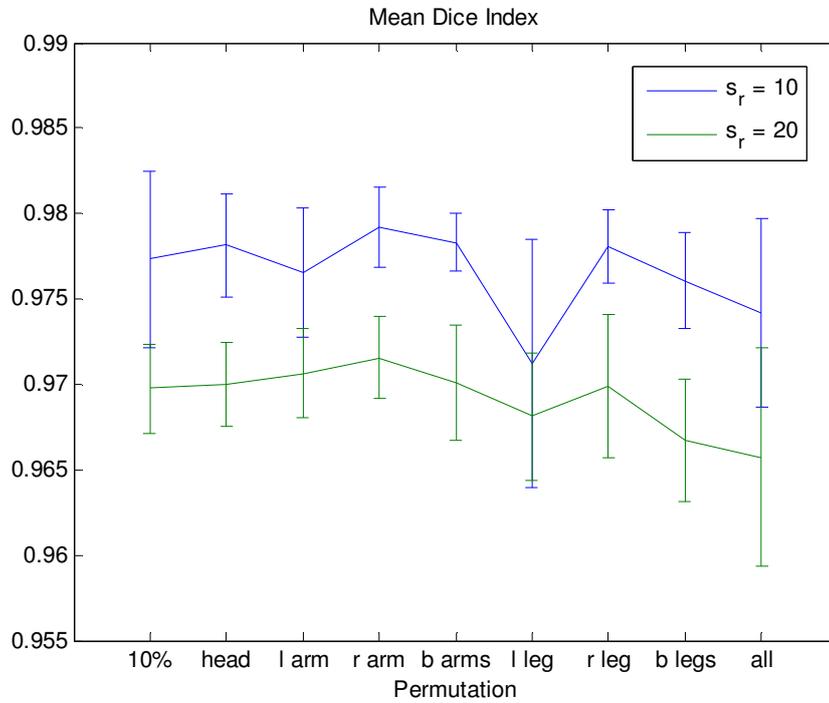


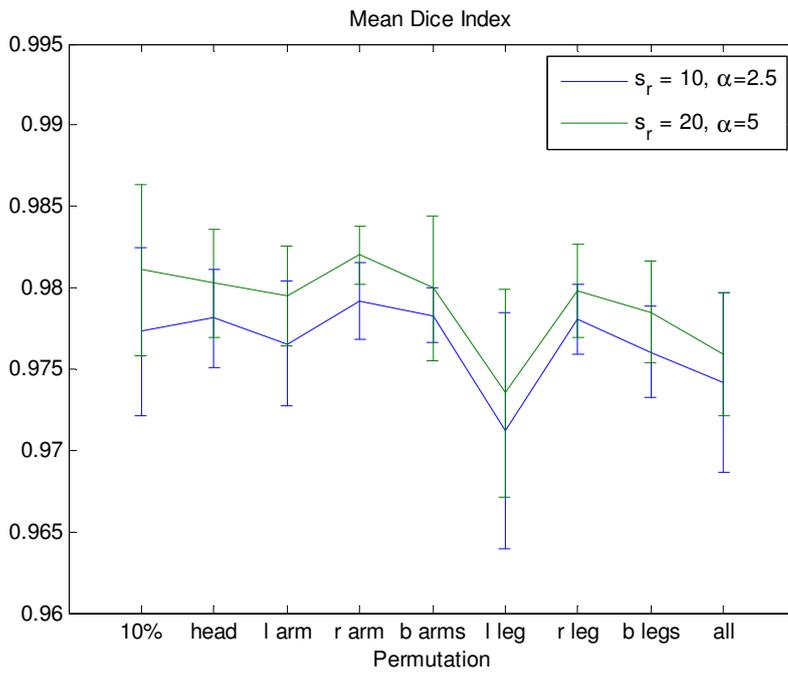
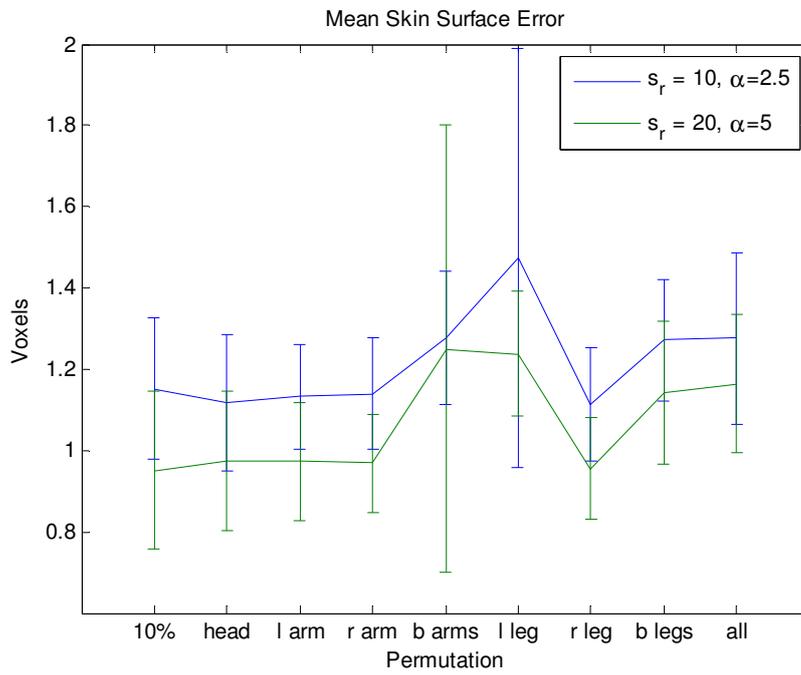
Figure 10: BLI registration

Changing the step size in the limb initialization (s_r in the paper)

When setting the s_r value to 10 instead of 20, both the mean skin surface error and the Dice index improve.



The α setting has a stronger effect than the s_r setting.



In Figure 11 and Figure 12, it can be seen that rotation errors in the elbow, also causes a rotation error in the wrists.

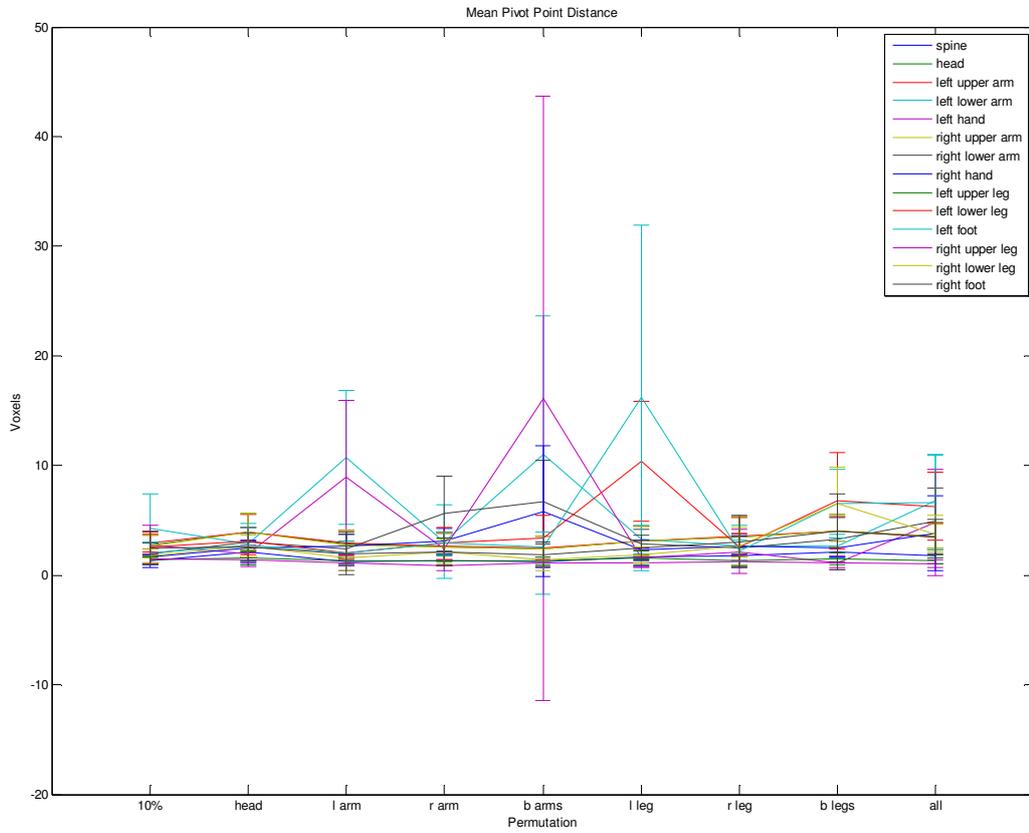


Figure 11: Pivot point distance errors of all individual rigid bodies.

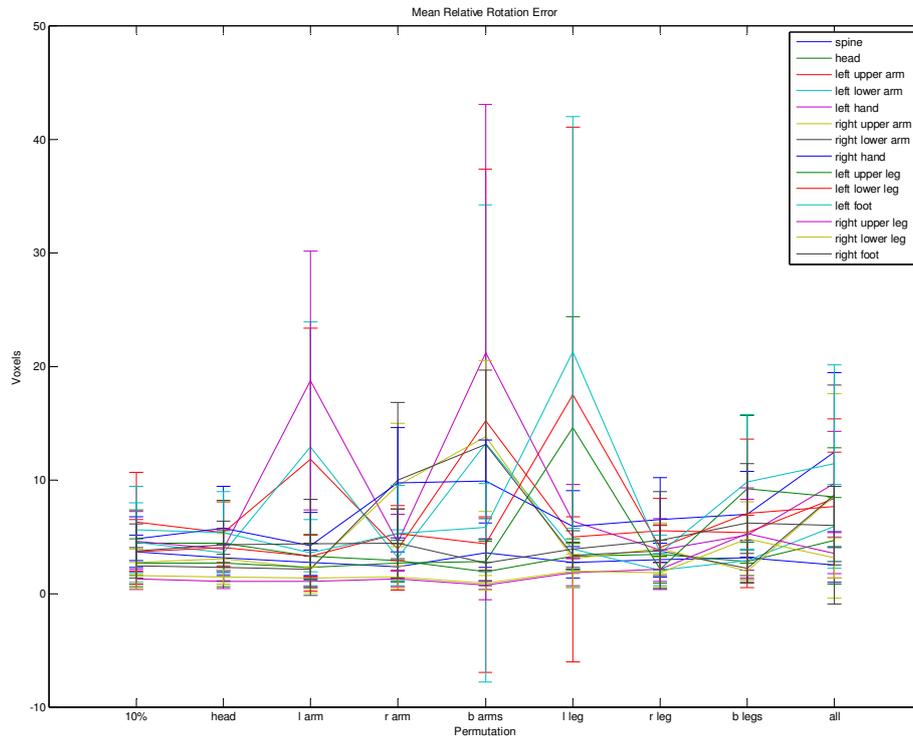


Figure 12: All rotation errors of each individual rigid body

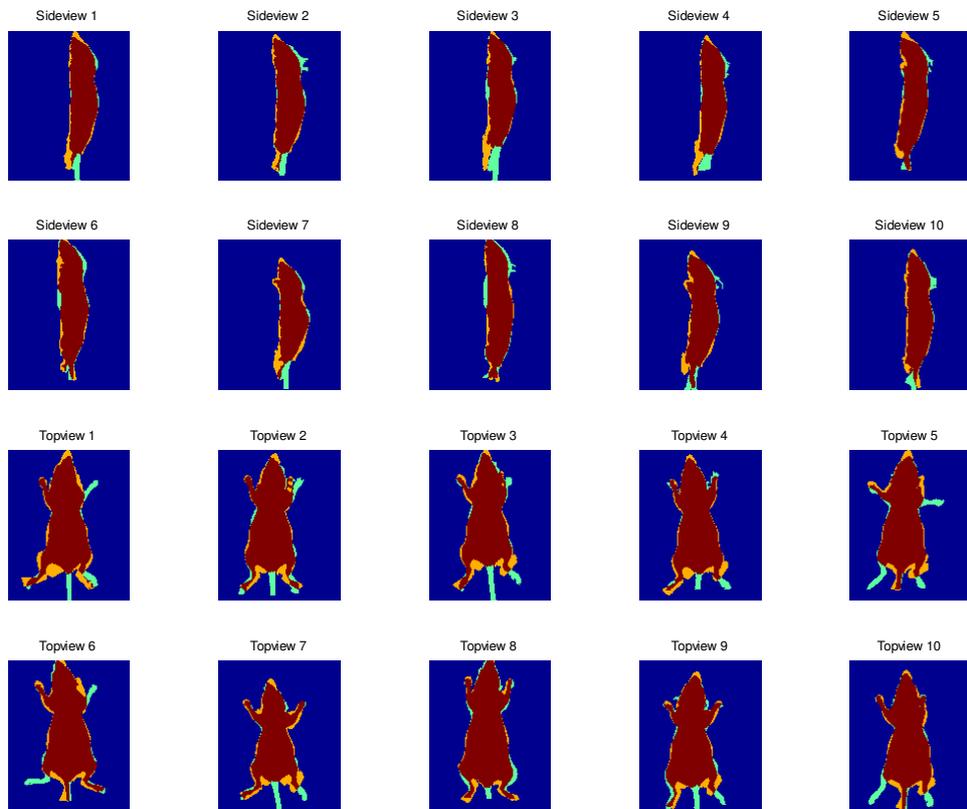


Figure 13: Projections of CT data and Mouse Atlas Registration.

The search for α :

A more extensive search indicates that an optimal α would lie somewhere at 5. In any case it is shown that an $\alpha > 1$ gives a better result on average than an $\alpha \leq 1$. Variance is almost zero, for the ten cases that we studied.

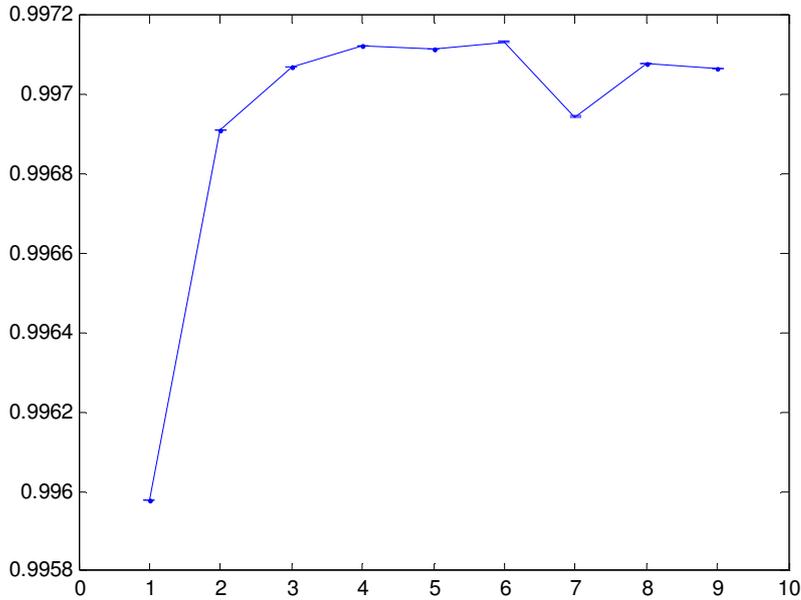


Figure 14: Dice's index vs. α for four views, resolution of 1

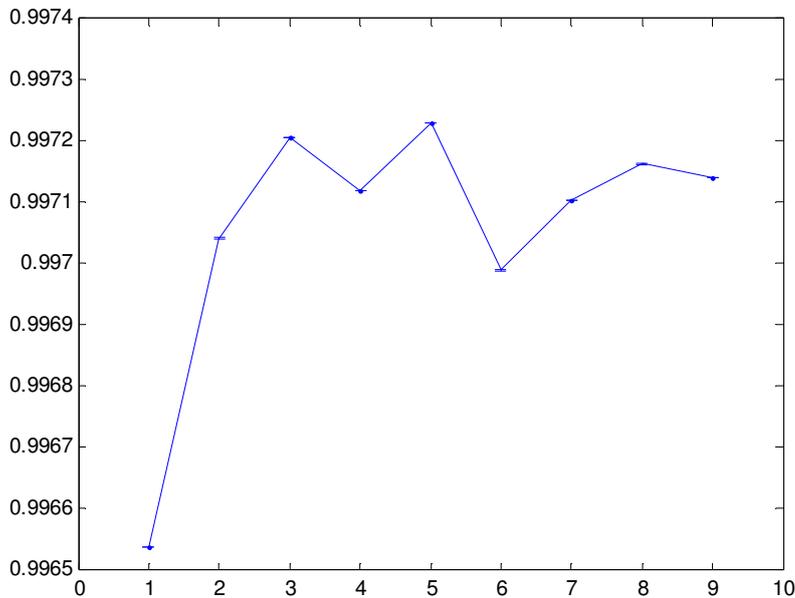


Figure 15: Dice's index vs. α for two views, resolution of 1

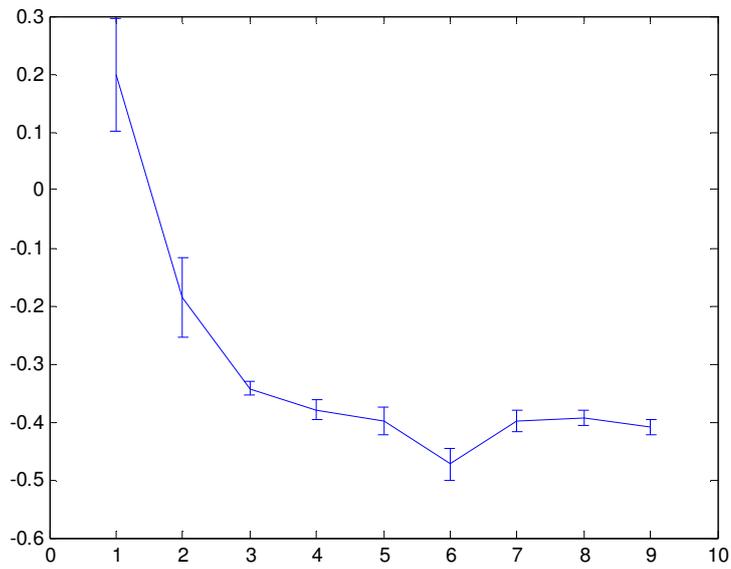


Figure 16: Scaling error vs. α for 2 view registration.

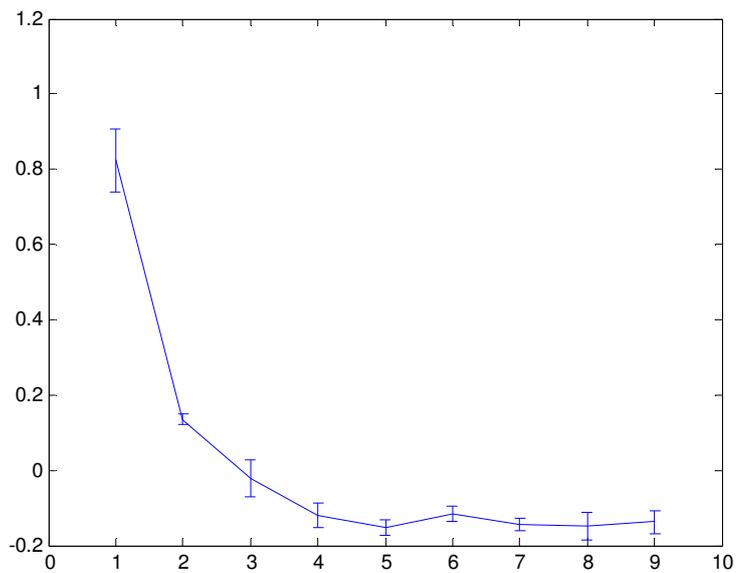


Figure 17: Scaling error vs. α for 4 view registration.

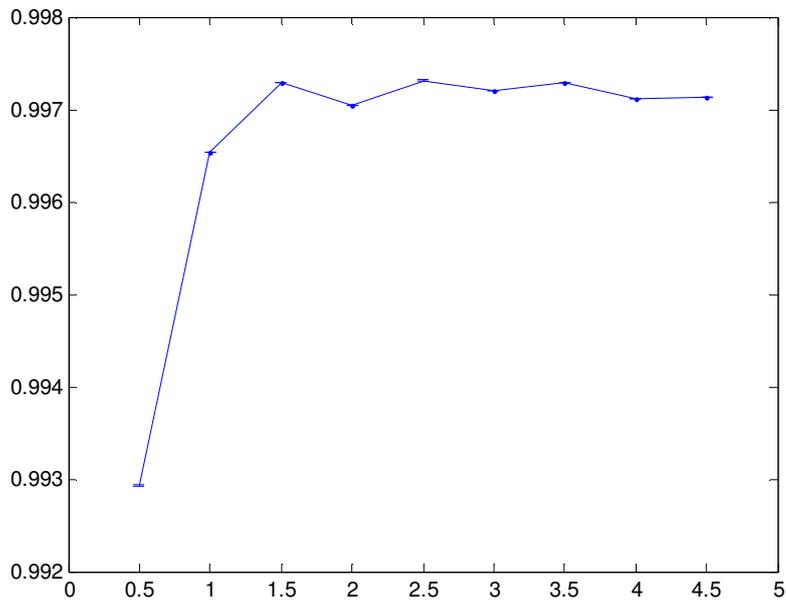


Figure 18: Errorbar of dice index vs. alpha (2view)

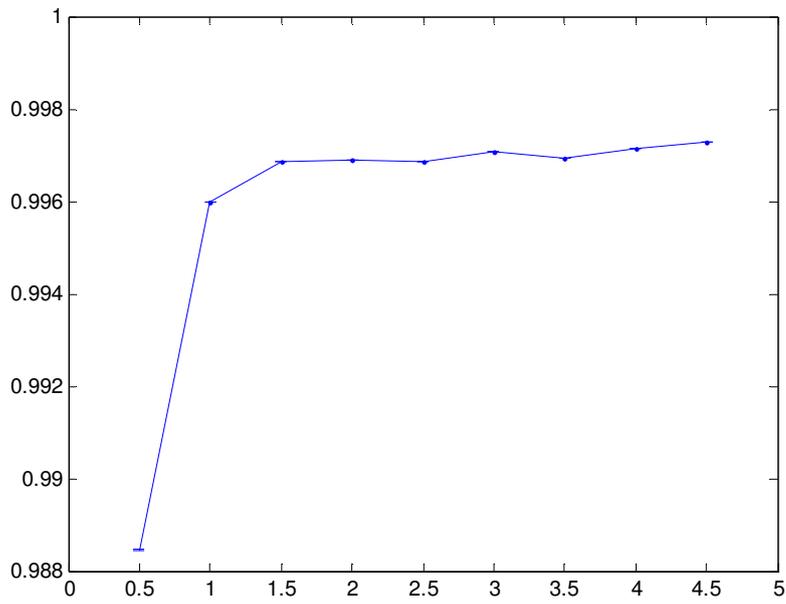


Figure 19: Errorbar of Dice index vs. alpha (4view)

Global Algorithm Currently Used:

- Read out top view
- Read out side view
- Combine top and side view into voxel model by multiplying
- Calculate distance transforms resulting in 3D distance matrix.
- Calculate thresholded distance transform, using 5 voxels.
- Calculate gradient directions resulting in three 3D matrices (for x,y,z direction), based on complete unthresholded distance transform.
- Normalize gradient-directions to length 1, so that they can be used for angle calculation
- Read out skin surface.
- Make global initialization by making sure the skin surface falls into the distance-matrix
- Start loop, making the allowed angle between gradient direction and skin surface normal, smaller each time.
 - o If allowed angle = 180
 - Use full distance matrix
 - o Else
 - Use thresholded distance matrix
 - o End if
 - o Calculate angle between vectors by taking the dot product
 - o If angle allowed
 - Read out distance transform matrix
 - o Else
 - Set value to maximum value that can be found in the distance matrix
 - o End if
 - o Return vector of distances
- Loop
- Plot figures

Mouse Manipulation Algorithm Used:

- Import Amira ascii digimouse surfaces into Matlab
- Import Manually Selected Pivot Points, including a point on x-axis and y-axis.
- Correct y-point so that it is orthogonal to x.
- Convert x and y points to unit vectors
- Calculate z unit vector by taking cross product
- Calculate Euler Angles that are needed to rotate Global Axis into Local axis. (Full optimization search)
- Merge all skeleton surfaces.
- Link each vertex on the skin surface to the nearest skeleton surface.

Do Scaling:

For each bone

- Map single bones to their origins
- Do inverse rotation so that global and local axis are the same, by using pre calculated Euler Angles
- Scale along x, y and z axis
- Rotate again with pre calculated Angles
- Map single bones back to original location
- Calculate the translation effect on children

Do Rotations:

For Each bone

- Calculate quaternions for x, y and z vectors
- Map bone and its children to origin
- Apply rotation on bone and on its children
- Apply rotations on pivots of children of bone (not on pivot of bone itself)
- Map bones back to original position

Do Transformations:

- For each bone, do translation. Also for pivots.

Additional Settings and Results.

	Min Scaling	Max Scaling				
All	0,8 * initial	1,2 * initial				
	Min Rot x	Min Rot y	Min Rot y	Max Rot x	Max Rot y	Max Rot z
Spine	-20	150	-30	20	210	30
Head	-5	-10	-20	5	10	20
Left Upper Arm	-10	-10	-40	10	30	60
Left Lower Arm	0	0	-20	1	1	90
Left Hand	-5	-5	-20	5	5	20
Right Upper Arm	-10	-10	-40	30	30	60
Right Lower Arm	0	0	-5	1	1	90
Right Hand	-5	-5	-20	5	5	20
Left Upper Leg	0	0	-40	1	50	40
Left Lower Leg	0	0	-30	1	1	80
Left Foot	-5	-5	-25	5	5	110
Right Upper Leg	0	0	-40	1	50	40
Right Lower Leg	0	0	-30	1	1	80
Right Hand	-5	-5	0	5	5	125

	Skin 1	Skin 2	Skin 3
Mean Absolute error	5.173724e+000	5.637252e+000	5.071760e+000
Mean Error	1.413894e-001	4.176248e+000	1.176282e+000
MSE	4.223858e+001	4.736853e+001	4.007259e+001

Initializing ... done in 0.226732 sec
Rasterize points in patches to grid points ... done in 4.49101 sec
Fill volume ... done in 3.72622 sec
reinitializing left arm
reinitializing right arm
reinitializing left leg
reinitializing right leg
Average absolute error : 6.270567e+000
Average error : 2.585499e+000
MSE : 7.688429e+001
Initializing ... done in 0.148426 sec
Rasterize points in patches to grid points ... done in 5.18034 sec
Fill volume ... done in 4.04634 sec
reinitializing left arm
reinitializing right arm
reinitializing left leg
reinitializing right leg
Average absolute error : 3.918637e+000
Average error : 2.633346e+000
MSE : 3.393262e+001
Initializing ... done in 0.159021 sec
Rasterize points in patches to grid points ... done in 4.20455 sec
Fill volume ... done in 2.66367 sec
reinitializing left arm
reinitializing right arm
reinitializing left leg
reinitializing right leg
Average absolute error : 5.935442e+000
Average error : 8.131472e-001
MSE : 5.618114e+001

Run : 1

Rotation Error in deg

	'x'	'y'	'z'
'spine'	[14.8438]	[12.9031]	[3.0051]
'head'	[1.7238]	[0.7753]	[0.6523]
'left upper arm'	[8.3314]	[9.2876]	[54.3048]
'left lower arm'	[0]	[0]	[4.2027]
'left hand'	[3.5800]	[3.2837]	[14.7493]
'right upper arm'	[9.2102]	[4.0728]	[25.5663]
'right lower arm'	[0]	[0]	[43.0534]
'right hand'	[4.7828]	[3.9240]	[15.9900]
'left upper leg'	[4.1170e-014]	[3.3998e-014]	[2.3065]
'left lower leg'	[0]	[0]	[29.9899]
'left foot'	[4.1618]	[2.7574]	[1.1409]
'right upper leg'	[0.5259]	[5.2538]	[19.2053]
'right lower leg'	[0]	[0]	[3.1323]
'right foot'	[0.2272]	[2.9298]	[3.1468]

Scaling Error in %

3.9959	6.8656	5.2022
0.7027	0.7027	0.7027
4.0186	4.0186	4.0186
4.0186	4.0186	4.0186
4.0186	4.0186	4.0186
10.2681	10.2681	10.2681
10.2681	10.2681	10.2681
10.2681	10.2681	10.2681
4.2914	4.2914	4.2914
4.2914	4.2914	4.2914
4.2914	4.2914	4.2914
2.3870	2.3870	2.3870
2.3870	2.3870	2.3870
2.3870	2.3870	2.3870

Topview area : 15588
Non covered area Top : 1124 (7.21 %)
Falsely covered area Top : 2054 (13.18 %)
Sideview area : 10480
Non covered area Side : 696 (6.64 %)
Falsely covered area Side : 668 (6.37 %)

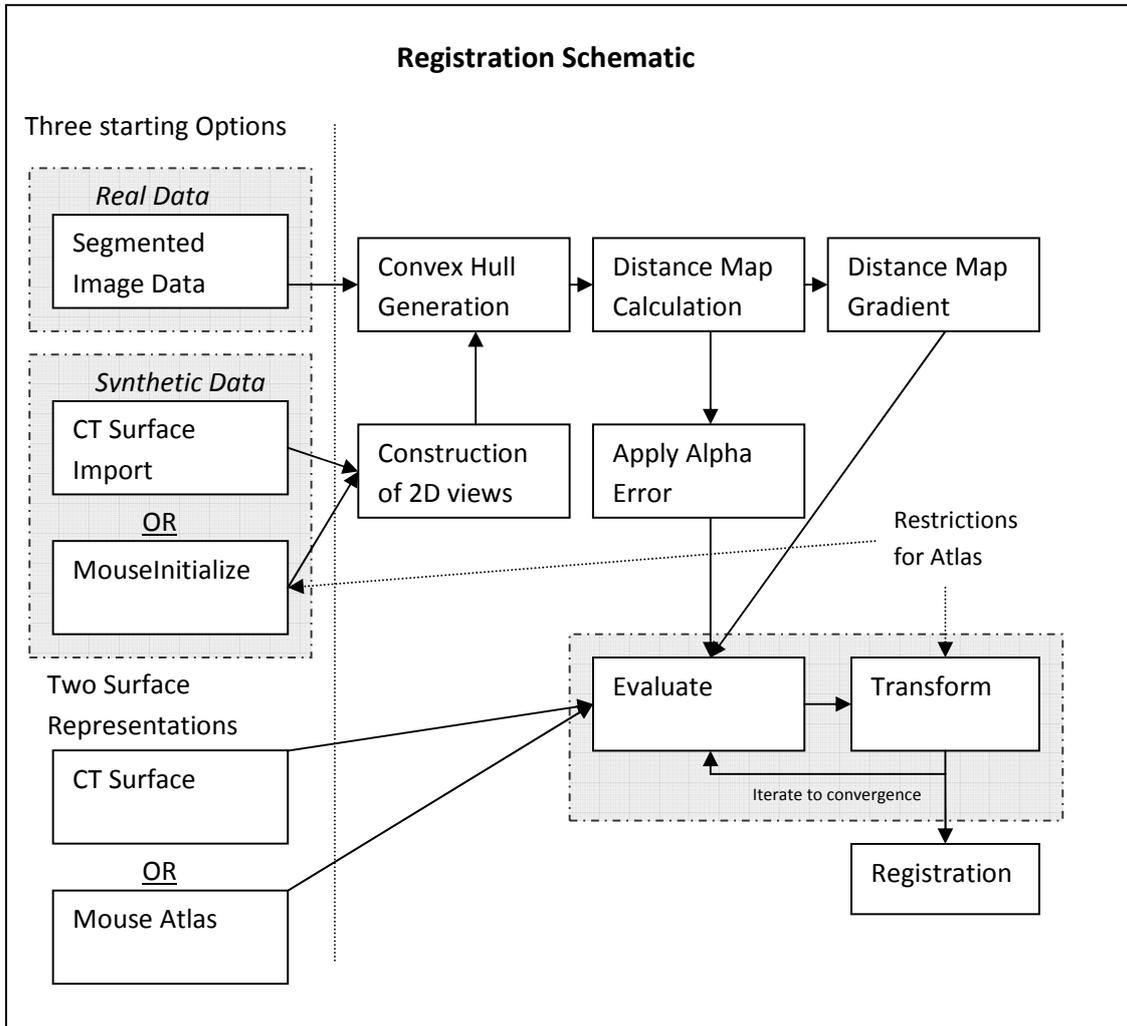


Figure 20: Schematic overview of the program. The Evaluate and Transform iterations (Gray area) differ for non rigid and rigid registration.

APPENDIX D

Working Document on *C. Elegans* Paper

Working Document on C. Elegans chronograms

On obtaining data

Problem: The chronogram dataset is only available in XML format.

Solution: Download the XML datafile and convert this XML data into matlab structures.

Problem: The server hosting the datafile crashes during download of the complete dataset.

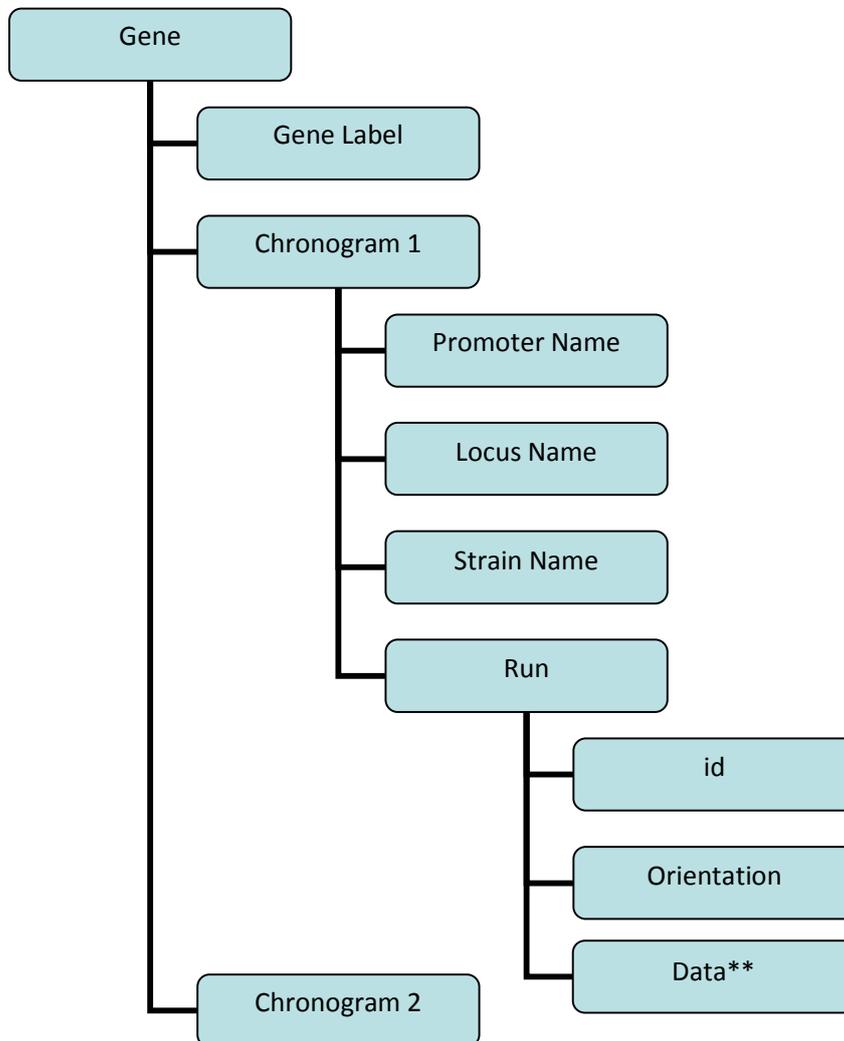
Solution: Download an 'empty' dataset, that only contains the names of genes the have a chronogram attached to it. With this file, download specific queries for this genes:

There are 1606 genes available, containing at least 1 chronogram per gene.

There are 2058 chronograms containing spatial temporal gene expression data.

There are currently three matlab structures.

A geneList structure, pointing to mat files, which in turn contain a structure of following format:



** Data contains the following fields:

wormgroupid: [500x1 uint32]
 orientation: [500x1 int8]
 datacount: [500x1 uint8]
 mean: [500x500 uint16]
 min: [500x500 uint16]
 max: [500x500 uint16]
 stddev: [500x500 uint16]

The localizome database

The following data is stored in the localizome database

Gene(s)	1606
Promoter(s)	1681
Strain(s)	1991
Chronogram(s)	2085

The meaning of a chronogram.

A chronogram is a representation of measurements of Promoter::GFP constructs. Each promoter is related to one genes and a gene can in theory have multiple promoters. Since the chronogram data reflects the observation of promoter regulation, the most logical combination would be to connects chronograms to promoters.

We therefore added all measured profiles, belonging to the same promoter, into one chronogram.

Since the raw data from the Localizome website isn't really raw anymore, (Alignment of different wormsizes is already done) we expected that global orientation was already resolved.

After comparing the images generated from data with the images generated by dupuy, some chronograms seemed to be flipped. We didn't correct for this yet, and as a result, the average profiles all looked alike.

All gene profiles are annotated with an orientation, except for gene **T18D3.4**, connotated with promoter **p_myo-2_AZ217_CGC**, where strain **AZ217** did not have an orientation in it's data tags. We compared this dataset with a thumbnail generated by Dupuy et al. and decided that it's orientation should be one. (i.e. its orientation was correct in the database.) During the processing of all chronograms, we saw some alignments that weren't internally consistent. i.e. not all heads were pointing in the same direction. This will make calculation from enrichment impossible for those cases.

Normalization of chronograms.

All chronochrams are normalized to a size of 401x450 datapoints. All datapoints that do not fall into this region are discarded. Dupuy et al. made measurements until most datapoints from 50 to 450 were observed. Chronograms need to be normalized to be able to compare them.

When normalized to surface of 1, you get a distribution of gene expression

2 Assumptions are made:

- Organlocation Changes proportional to size of worm duing development
- Gene Expression is not age related.

Selecting Statistical Relevant Signals.

As was described in supplementary materials by Dupuy, we generated a background signal (B) with the 68 lowest mean signal chronograms. We kept 1531 chronograms, instead of the 1520 that was reported in Dupuy. (The reason they used the number 68 is probably fully arbitrary an generated best results) We calculated a mean chronogram and a standard deviation , $m(B)$ and $\sigma(B)$

after this we calculated the Z score according to the formula, as was shown in Dupuy:

$$Z = \frac{\sum_i^n C_i - m(B)_i}{\sigma(B)_i} \text{ where } i \text{ is datapoint location, after a chronogram was flattened. (i.e.}$$

instead of stacked on top of each other, all worms were concatenated.

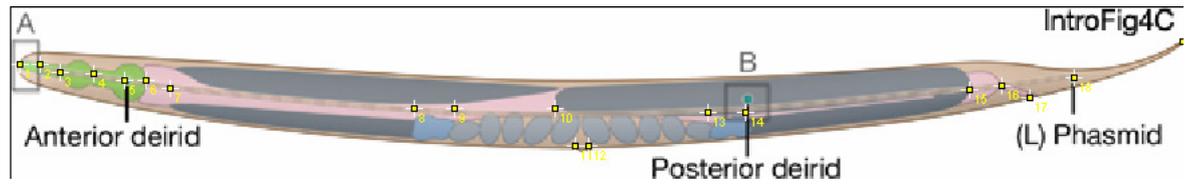
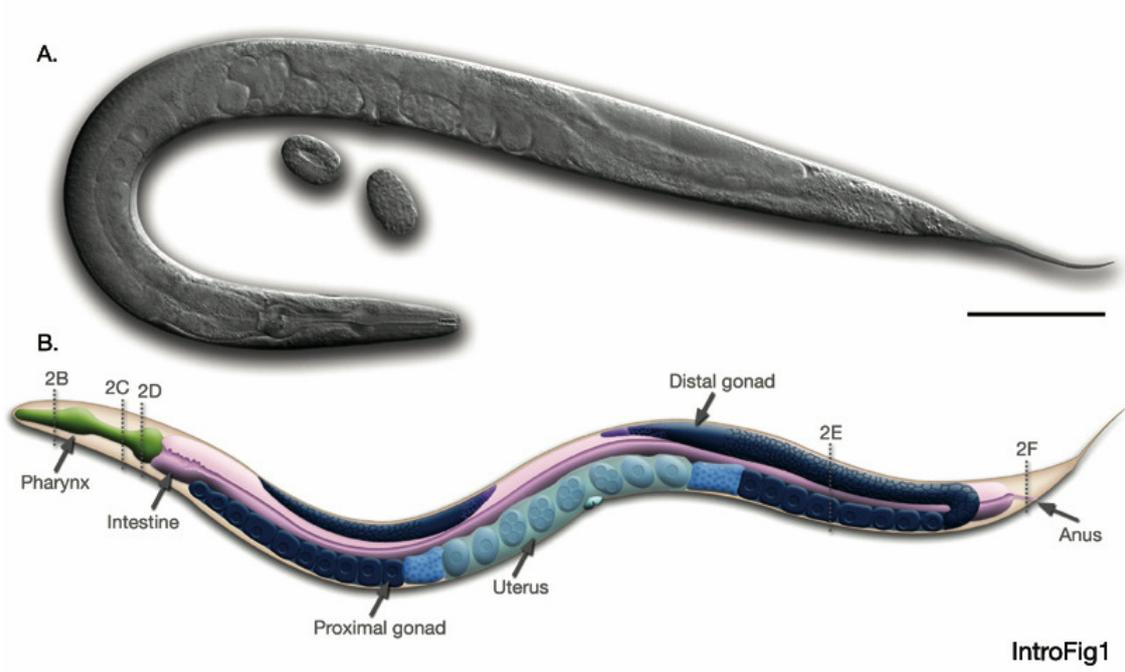
Instead of the reported 820 chronograms having a Z score of 11 or more. (Again, it is unclear where this value of 11 originates from) we found a total of 94 chronograms.

Constructing the Worm Atlas.

There are two ways of constructing a wormatlas.

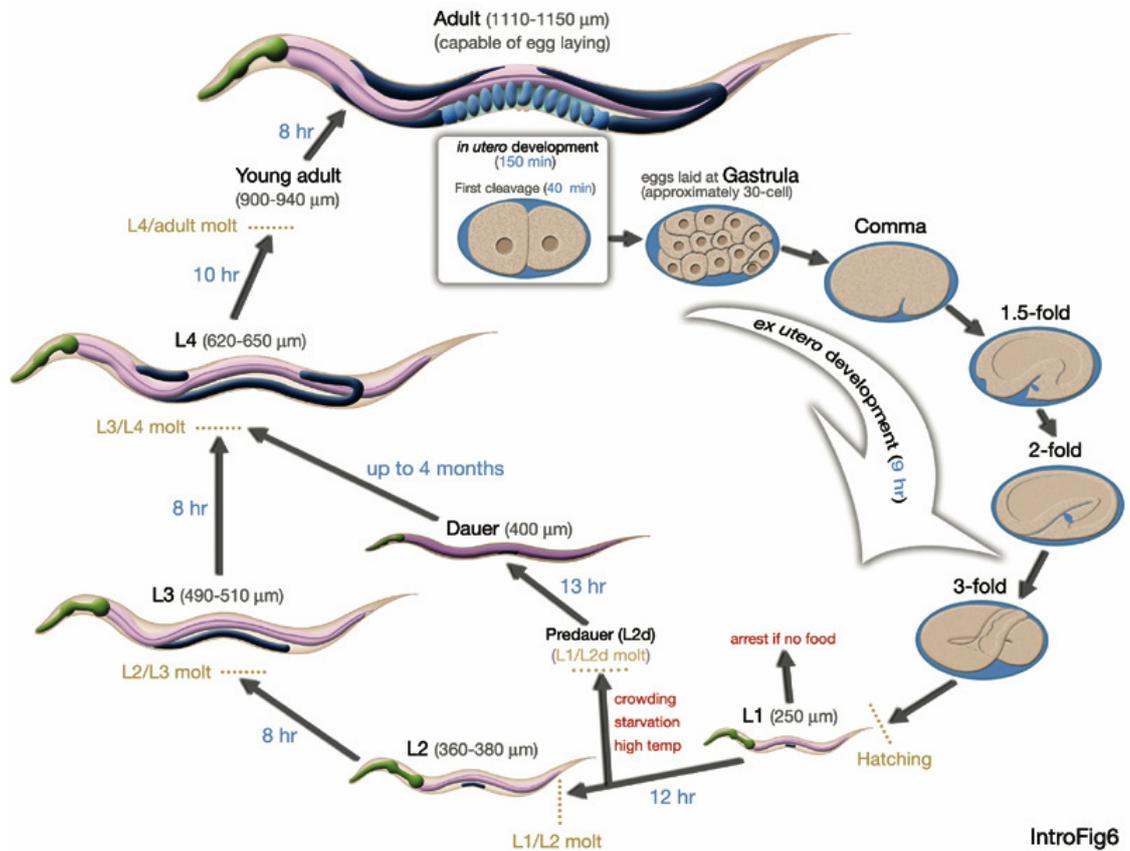
First, we can construct an atlas by using known anatomical location found in literature. Since we have a 1D worm structure, all 3D anatomical information that we find have to be translated to a 1D (binary) model.

Secondly, we can make use of chronogram averages and take the highest observed peaks in the chronograms, to construct an atlas, using a data driven approach.



Marked points from anterior to posterior.

1	4
2	14
3	24
4	41
5	56
6	67
7	79
8	200
9	220
10	270
11	280
12	287
13	346
14	365
15	476
16	492
17	506
18	528
19	582



IntroFig6

Figure 1: Different stages in *C. Elegans* development
 (<http://www.wormatlas.org/handbook/anatomyintro/anatomyintro.htm>)

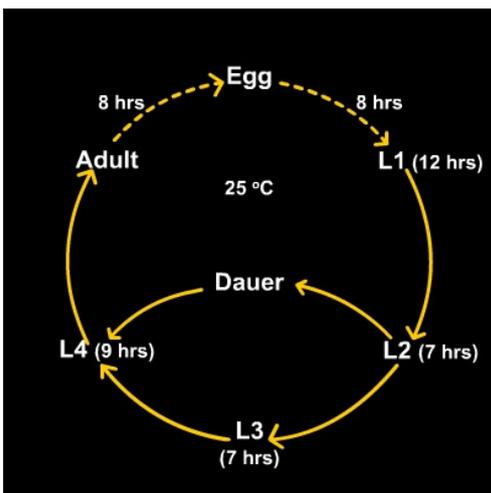


Figure 2: Lifecycle (from <http://www.wormclassroom.org/ac/lifeCycle.html>)

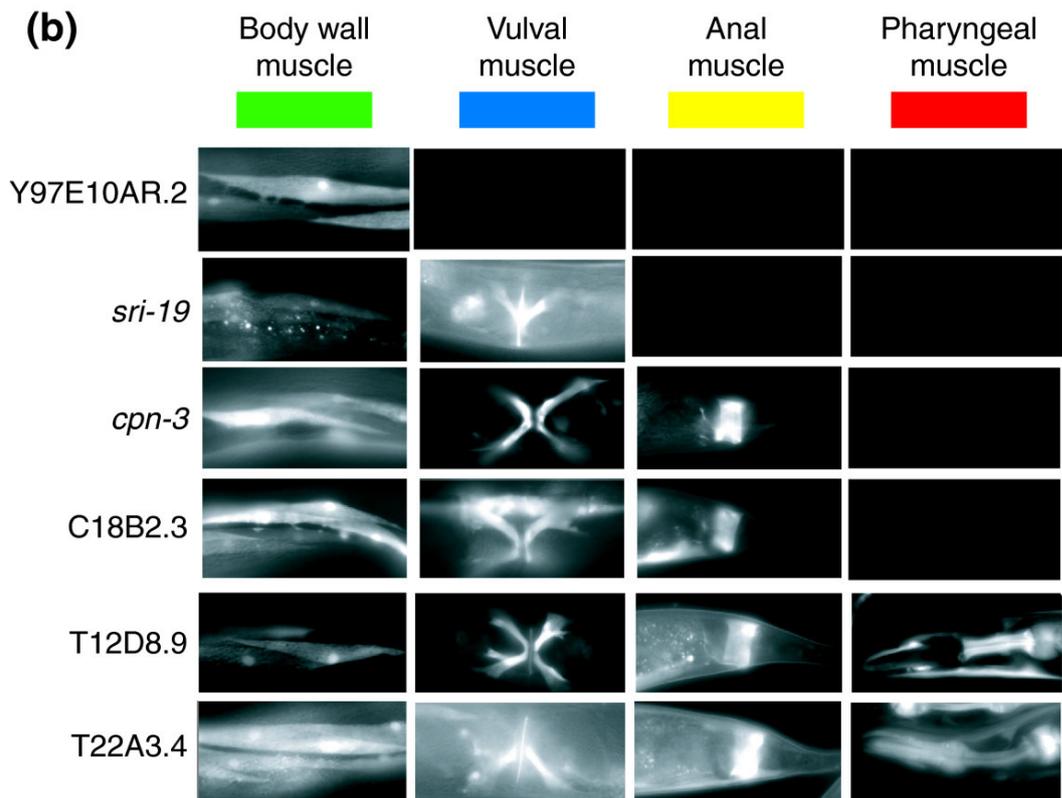
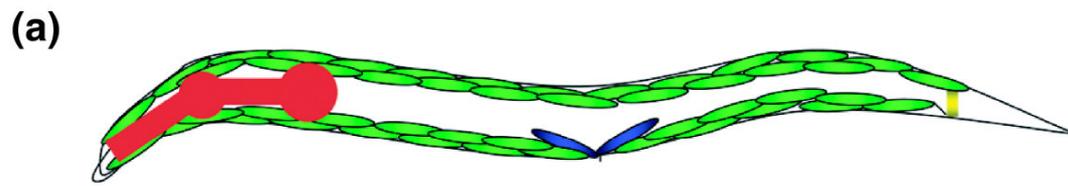


Figure 3: From The embryonic muscle transcriptome of *Caenorhabditis elegans* (Rebecca M Fox)

Determining the correlation between chronograms.

Of these values we calculated the Pearson's Correlation Coefficient:

$$R(i, j) = \frac{C(i, j)}{\sqrt{C(i, i)C(j, j)}} \text{ and } C(i, j) = \sum_x i_x j_x - \frac{\sum_x i_x \sum_x j_x}{n} \text{ where } i \text{ and } j \text{ are vectors of size } n$$

where i and j are two equally sized chronograms, and x is the datapoint index and n is the number of datapoints in i (or j)

Resulting in Figure 4. We expect the chronograms having the highest correlation, to be expressed in the same organs of the worm.

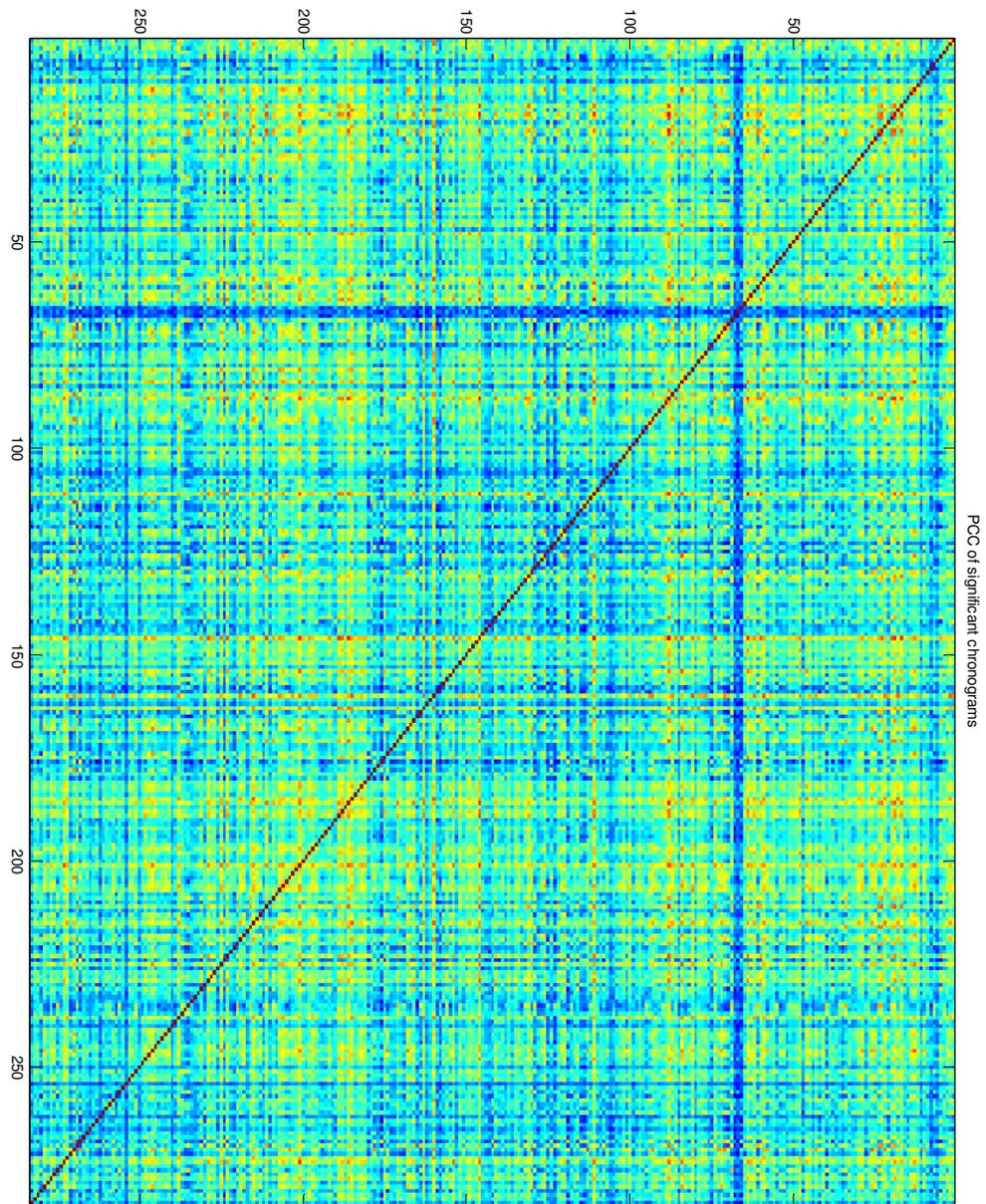


Figure 4: PCC of all chronograms vs. all chronograms. The subset of chronograms used, is the set with a Z-score > 11.

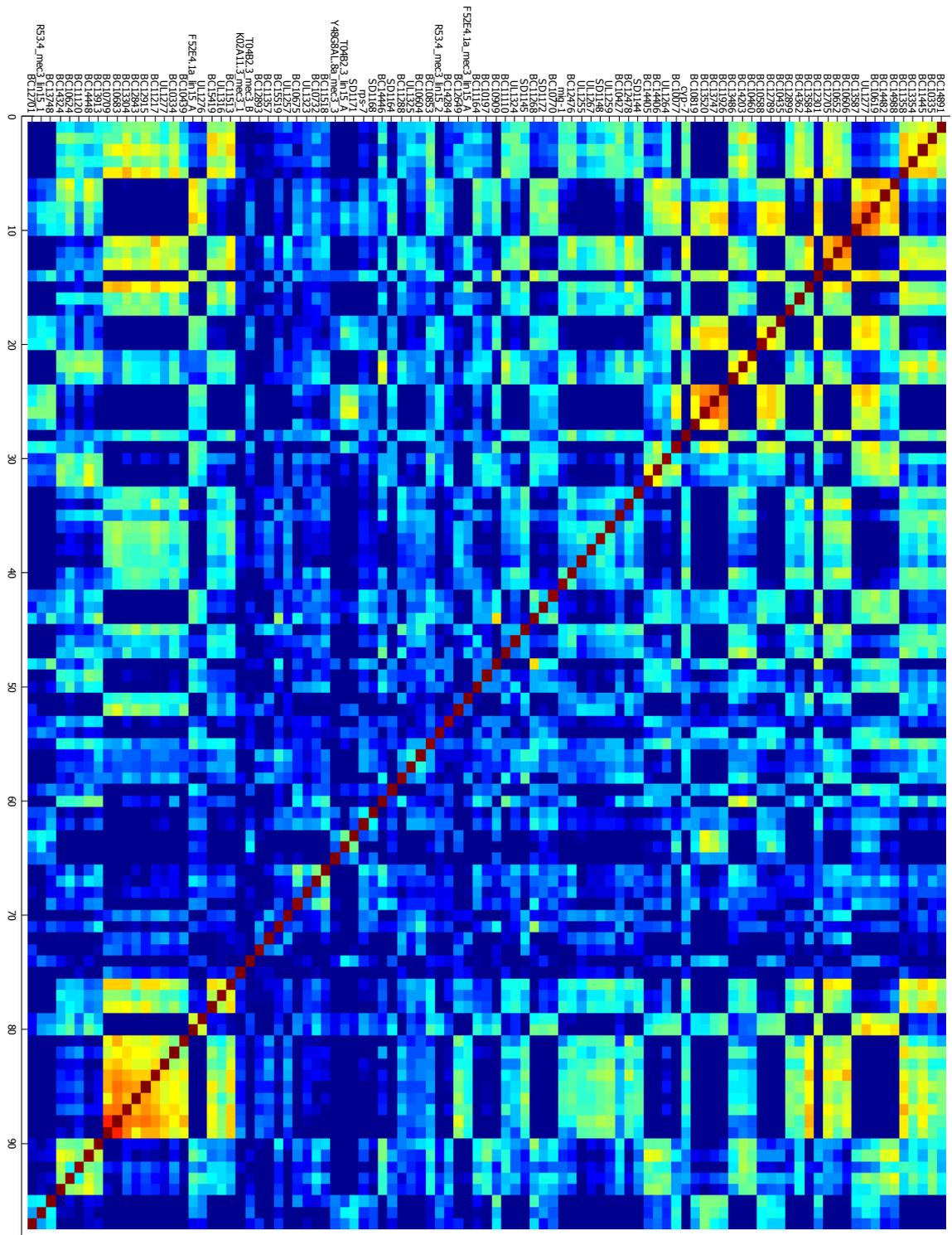
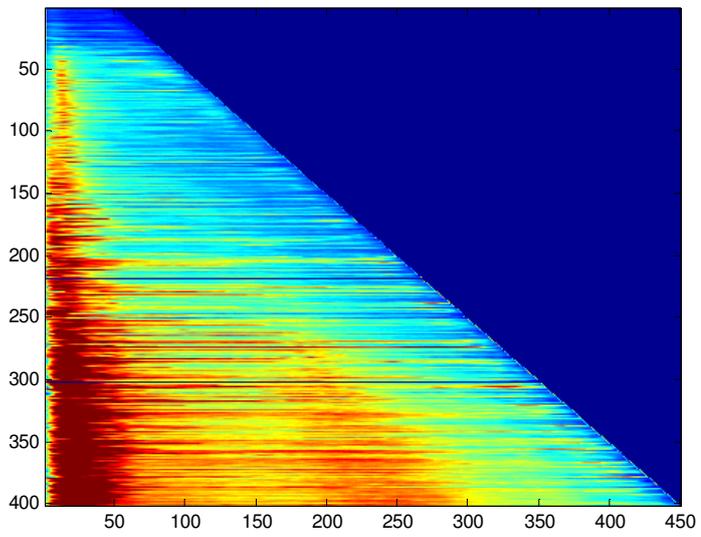
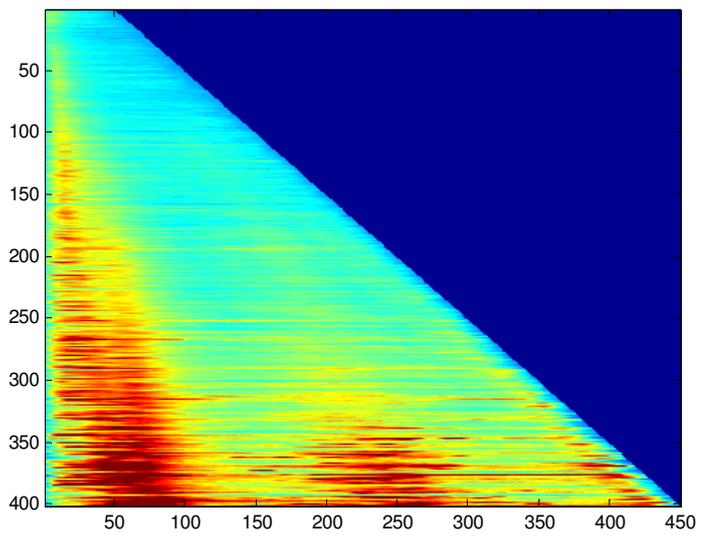


Figure 5: Clustering using Average Linkage

The highest High Pearson Correlation observed was based on following 2 chronograms.



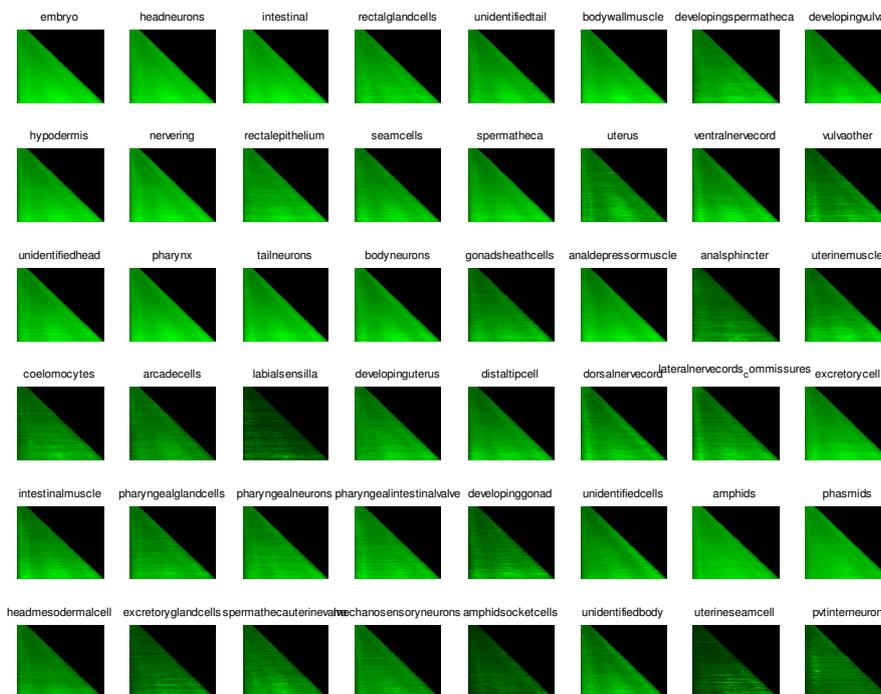
How to determine spatial expression profile:

I parsed anatomical information from the Localizome website. Each strain belongs to multiple anatomical expressions. This makes thing complicated.

We generated average chronograms by adding all expression profiles annotated with some anatomical profile together.

Hopefully correlation will show up higher for chronograms that have some related anatomical information.

A quick qualitative inspection unfortunately does not show large differences between different anatomical view.



As can be seen in the plots of the averaged chronograms, some averages are affected by datasets with too many missing datapoints (resulting in a lower mean intensity and some high intensity bars). We therefore applied a filter which removes all chronograms from the averages that had more than 60% of datapoints missing. This still seems like a lot of missing data, but it is in accordance with Dupuy et al. and they seemed to get good structural information when using this as filter.

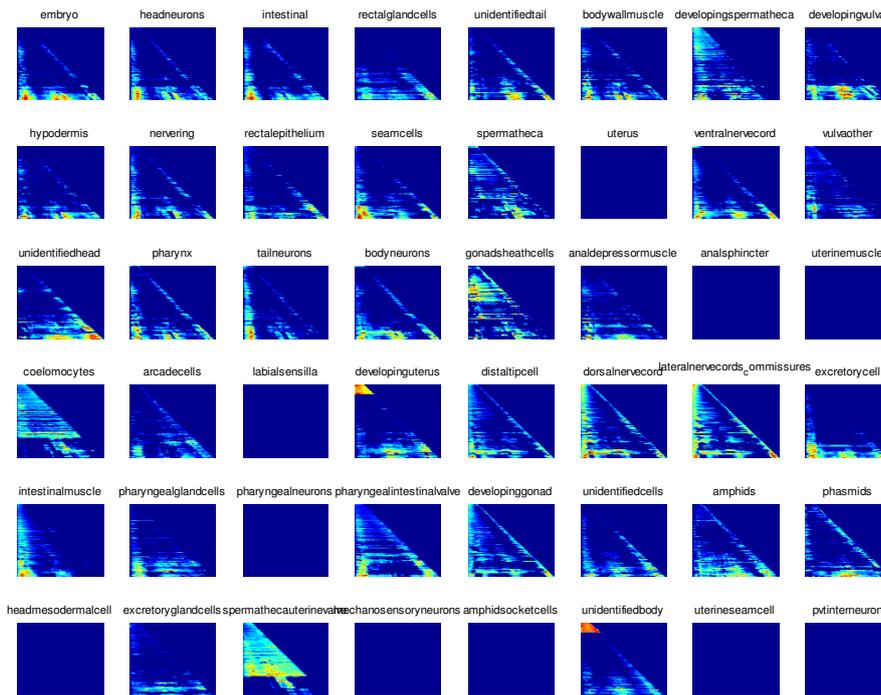
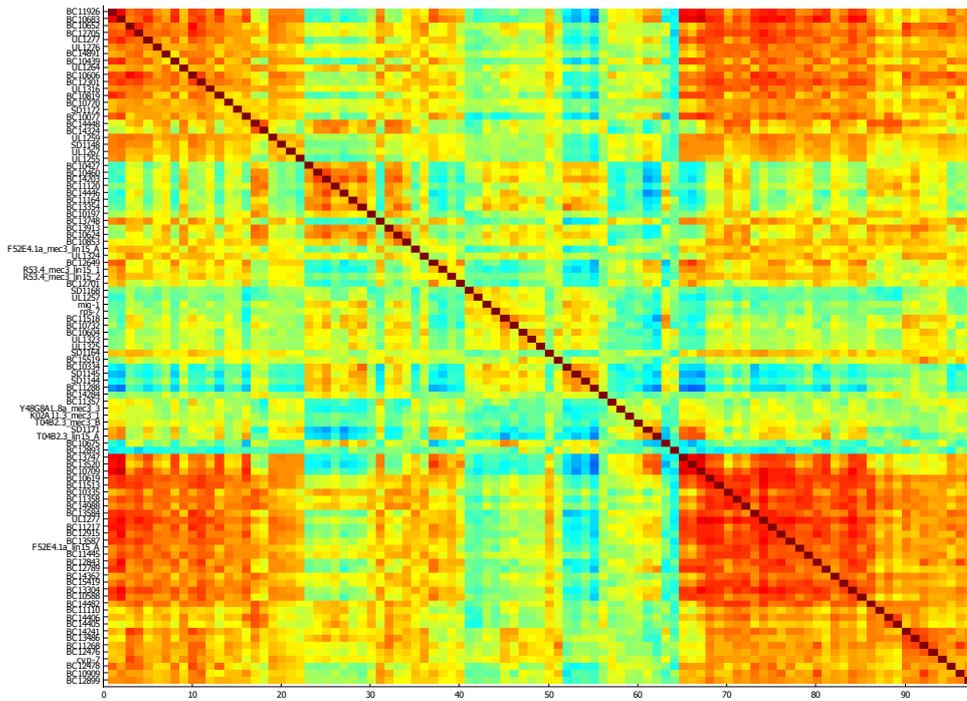
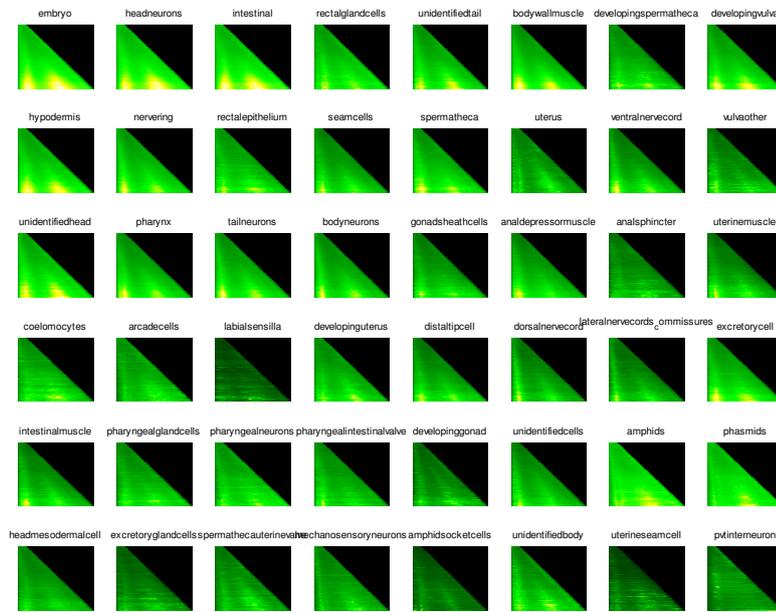


Figure 6; Difference of filtered vs non filtered

After correcting for the orientation of the chronogram data, we observe strong correlation and more over, larger clusters, which is a logical result.



We now also expect the anatomical ‘fingerprints’ to be more different. We do see difference in the ‘fingerprints’ but there is still no obvious difference between the anatomical annotations. The largest difference is that in stead of three regions of strong expression, observed in the uncorrected chronograms, we now see only two regions of strong expression.



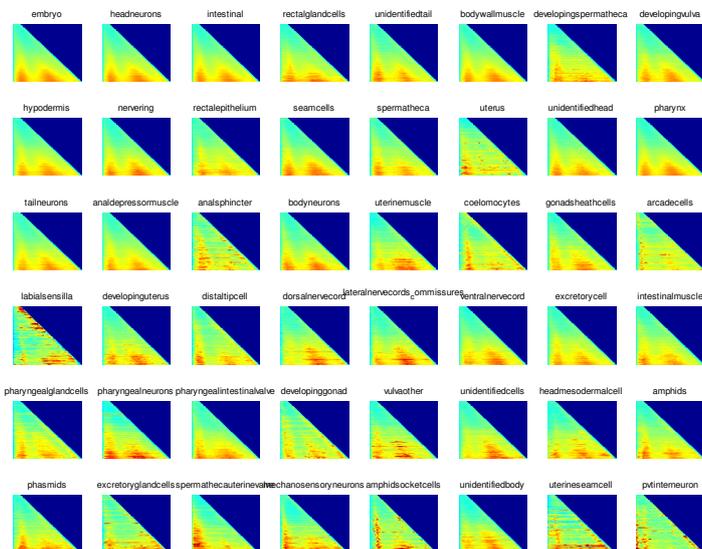
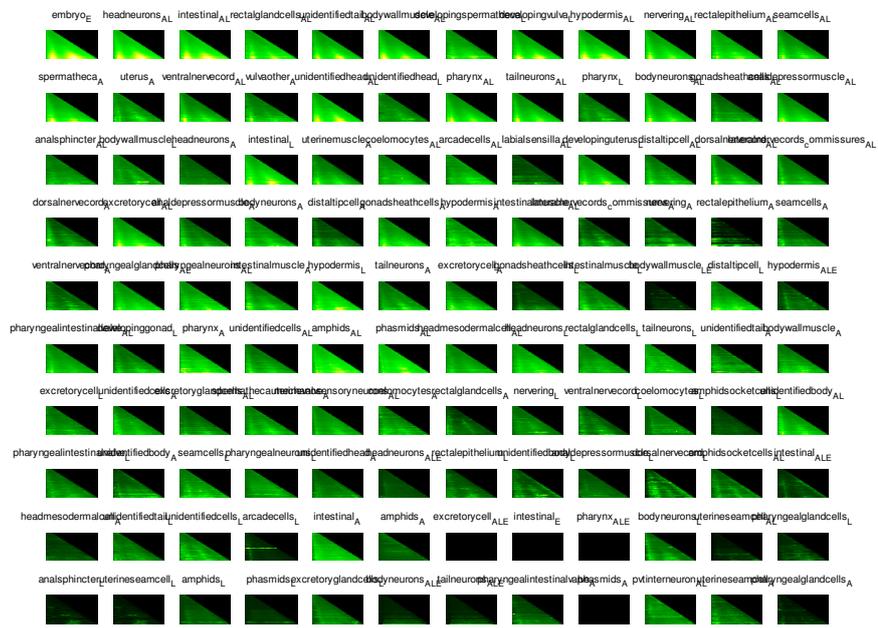
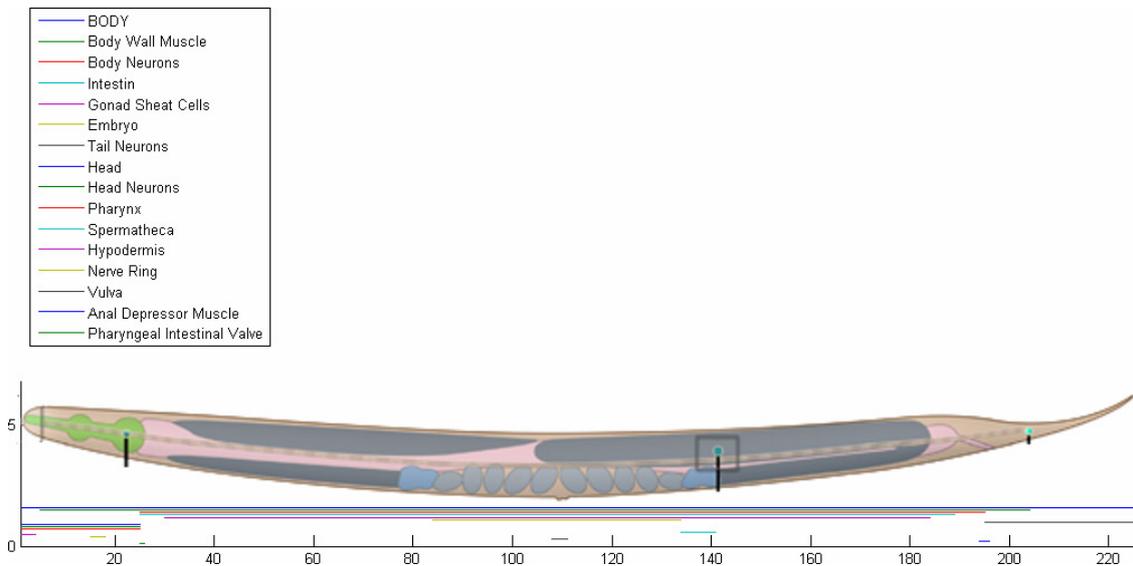


Figure 7: Final obtained fingerprints.

Update: We found a bug in the code, which caused an incorrect merging of strain chronograms. Everything has to be redone! All statistical tests and fingerprint calculation.

Since we do not find any obvious differences between the averaged chronograms having shared ontology tags, we decided to make use of the manually segmented atlas.

Since we have a 1D worm, we have many organs that overlap. This is not a problem when trying to answer whether a gene is 'enriched' in a certain area. When trying to answer in what organs the observed genes are expressed difficulties do arise.



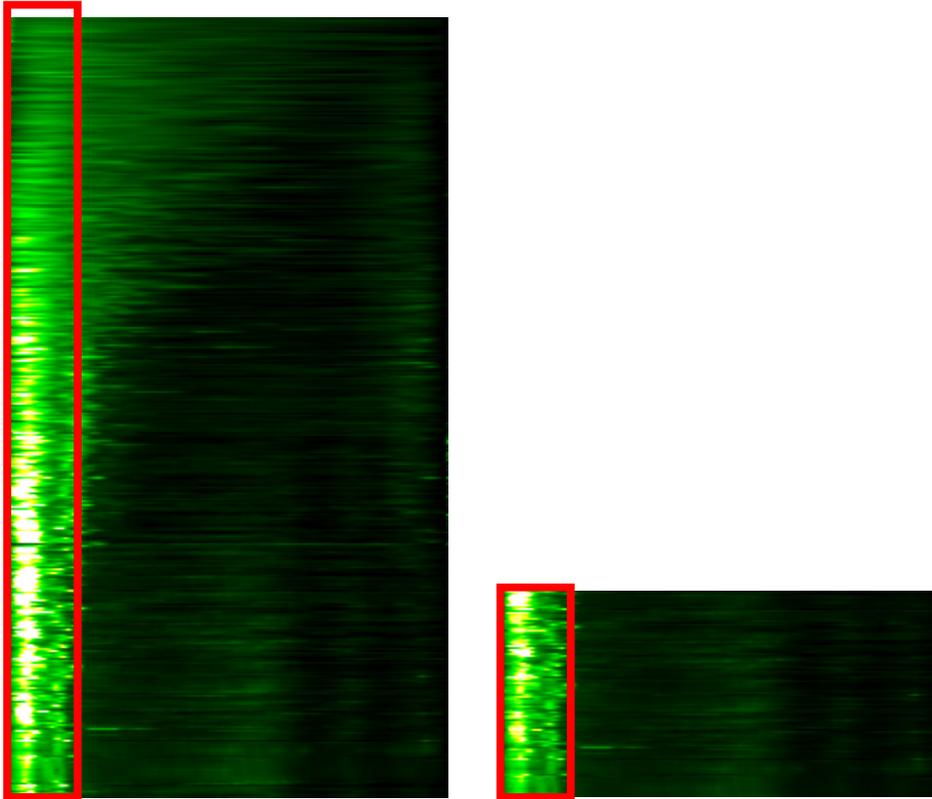


Figure 8: Chronogram of a p_H28O16.1_BC::GFP construct. Likely to be enriched in the head. Left the full chronogram, right the datapoints which we define to be in the adult stage.

To test for enrichment, a two sample t-test can be applied, where the gene expression samples are divided into two groups. In the organ area, or not. We can then test for different mean expression in one worm, but also over a complete chronogram, (or a subset of a chronogram).

Relation to age:

The atlas we constructed is based on adult worms. The used chronograms consist out of larval and adult stages. We have to determine when the transition between larva and adult occurs. This is not a discrete event and occurs somewhere between the size of 650 and 900 μm . The transition is triggered by *let-7*. (There is no chronogram of this gene available)

To be sure that our spatial atlas corresponds to observations of the adult phase, we only use 900 to 1150 μm as our data source. (Based on Figure 9)

In a chronogram 0.1mm equals 50 datapoints and ~ 1 mm probably equals 500 datapoints. This would mean that we can only use 450+ datapoint measurements, which leaves us with 1 worm

size per chronogram. Instead we use 0,65 mm+ as size which gives us the wormsize 325-450 to apply a t-test on.

We are confined to this approach, because we cannot recover the meaning (e.g. the size) of a datapoint. 325-450 seems a safe guess, since it looks like Dupuy et al mark 300+ as the adult stage. This would translate into line 276-401 in our chronograms (containing size 325-450)

When using a t-test the assumption is made that the observed data is drawn from a normal distribution.

When using a two sample t-test you are able to test the null hypothesis H_0 that the means of both datasets is equal. Since this difference is present in many cases, this might not be a good test for enrichment. See example below:

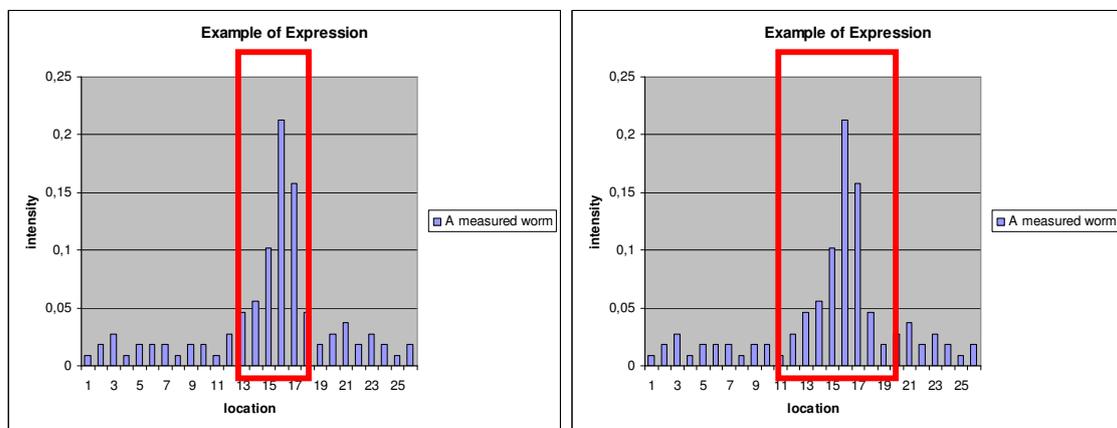


Figure 10: Selected regions based on atlas. Both regions will get a significant different mean, when applying a t-test

We also used a chi-square test to test whether the observed data subset selection seen in the frame can be explained by the normal distribution seen in the whole worm. To do this, the observed mean expression and its variance in the worm are calculated. After that the data observed in the selected region is tested against a null hypothesis “observed data is generated by a normal distribution with given variance”. The H_0 is rejected below a 5% significance level.

To make use of the multiple rows that we have, we applied the chi-square test to all row separately. We calculated a fraction of rejected H_0 and sorted the results of test from high to low rejection fraction. The largest fractions are differently expressed in most of the observed areas.

Problem:

The chronogram data does not seem to be normally distributed. Some random examples below (Figure 11):

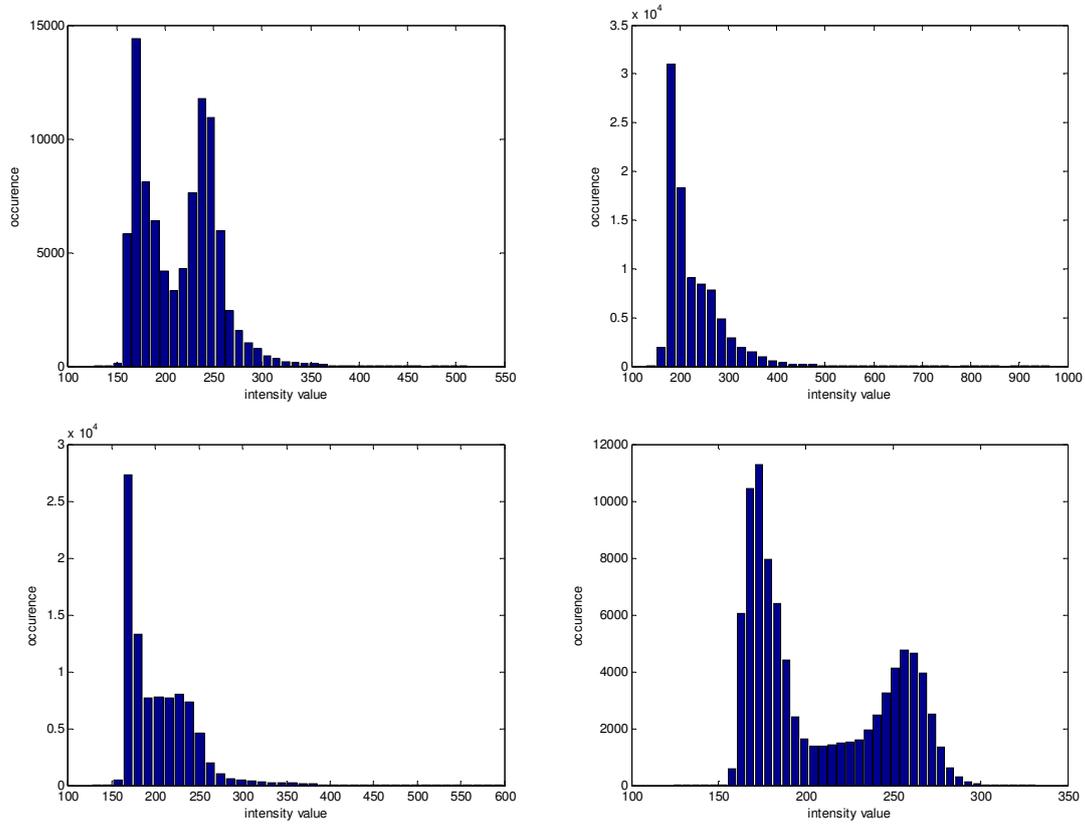


Figure 11: The distribution of intensity values of 4 Randomly picked chronograms

In fact, the distribution seems to be varying greatly and no assumptions can thus be made about the distribution of the data. It is thus not possible to perform a t-test, nor a chi-square test, because both of the tests assume normal distribution.

Based on the following table, it seems logical to pick a Wilcoxon test, to rank the scores. The same problem still remains though. With the Wilcoxon test, we are only able to test for a difference in distribution.

Table 1: Choose a test (From Intuitive Biostatistics, Chapter 37)

	Type of Data			
Goal	Measurement (from Gaussian Population)	Rank, Score, or Measurement (from Non- Gaussian Population)	Binomial (Two Possible Outcomes)	Survival Time
Describe one group	Mean, SD	Median, interquartile range	Proportion	Kaplan Meier survival curve
Compare one group to a hypothetical value	One-sample <i>t</i> test	Wilcoxon test	Chi-square or Binomial test**	
Compare two unpaired groups	Unpaired <i>t</i> test	Mann-Whitney test	Fisher's test (chi-square for large samples)	Log-rank test or Mantel-Haenszel*
Compare two paired groups	Paired <i>t</i> test	Wilcoxon test	McNemar's test	Conditional proportional hazards regression*
Compare three or more unmatched groups	One-way ANOVA	Kruskal-Wallis test	Chi-square test	Cox proportional hazard regression**
Compare three or more matched groups	Repeated-measures ANOVA	Friedman test	Cochrane Q**	Conditional proportional hazards regression**
Quantify association between two variables	Pearson correlation	Spearman correlation	Contingency coefficients**	
Predict value from another measured variable	Simple linear regression or Nonlinear regression	Nonparametric regression**	Simple logistic regression*	Cox proportional hazard regression*
Predict value from several measured or binomial variables	Multiple linear regression* or Multiple nonlinear regression**		Multiple logistic regression*	Cox proportional hazard regression*

Results of the chi_squared test:

For the genes, enriched for the tail genes, we found the following top10 registered chromograms:

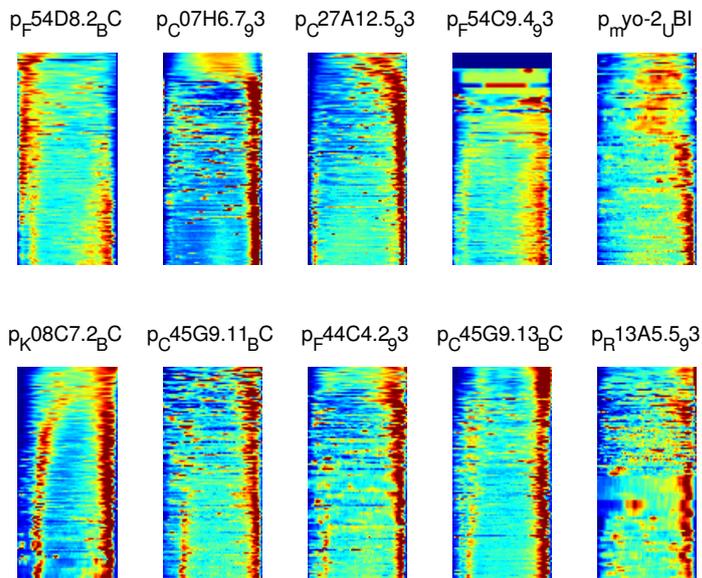


Figure 12: Tail enriched, found with chi_squared test

For the genes, enriched for the head, we found the following top10 registered chromograms.

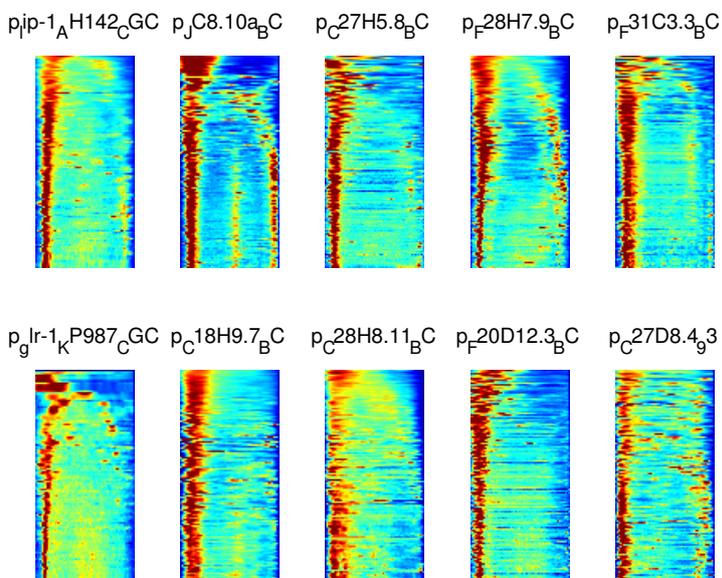
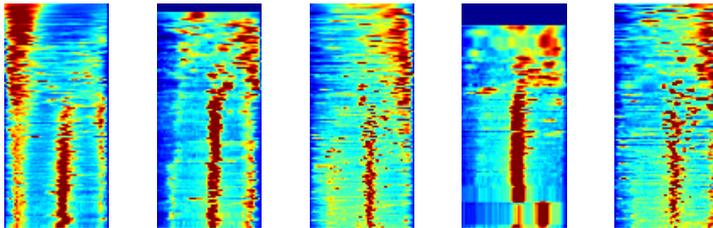


Figure 13: Head enriched, found with chi_squared test.

All, but one chronogram contained the annotation 'head' or 'pharynx'. p_F28H7.9_BC, which is expressed in the nerve ring and tail neurons, was falsely marked as 'enriched' in the head, but because of Figure 10, this makes sense.

Embryonic Development also showed some sensible results.

p_F42E11.2a_B₆gl-15_NH2447_CGCp_C06G4.4₉3p_c eh-24_pD4655_CGCp_C04G2.7₉3



p_F22B3.1_B_C p_F35G12.6_B_C p_F47H4.1₉3 p_F48F7.1_B_C p_K11D2.2_B_C

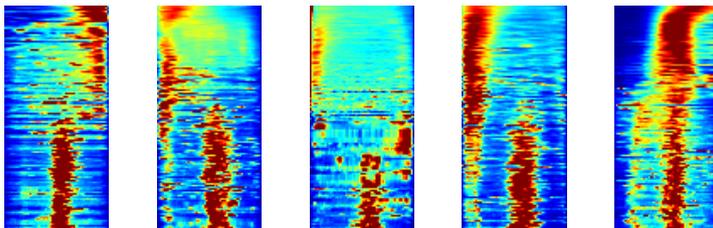


Figure 14: Embryonic Development, found with the chi-squared test.

Results of Wilcoxon's test:

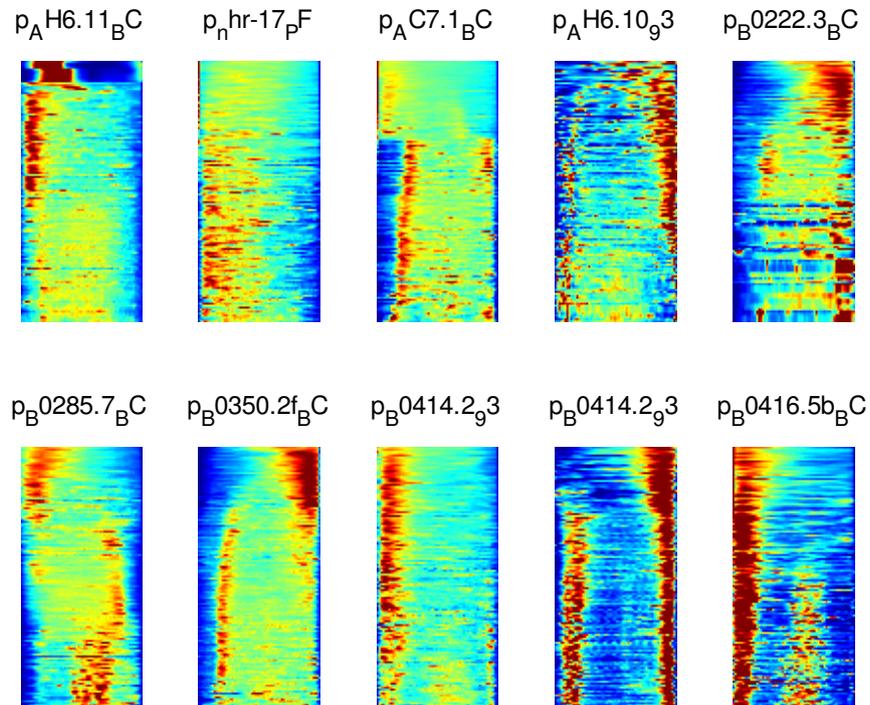


Figure 15: Enrichment of the head, Wilcoxon's test.

The results of the Wilcoxon's test were a bit disappointing. No pattern was discovered in the chronograms, where the other tests showed very clear enrichment.

Update(!): The implemented Wilcoxon test in matlab (ranksum) selects the smallest sample to be the first sample and the largest sample to be the second one. When using the z-Statistic as a measure for ranking, you will get an inverse order of what you expect, if your first sample is larger than your second one.

```

if nx <= ny
    smsample = x;
    lgsample = y;
    ns = nx;
else
    smsample = y;
    lgsample = x;
    ns = ny;
end

```

As workaround we sort ascending when or first sample is larger than 0,5*wormsize and descending otherwise.

Can't we just use correlation?

The chronogram that is shown in Figure 8 was found using a correlation estimation. While we only used the largest worm for this test (The worm with size of 450 datapoints).

Although this test is a very simple way to search for enrichment, we found that first experiments show good results. When we take more worm sizes into account, we can chose to make a score with more, or less weight in the weighted correlation score that follows.

When all sizes are weighted equally, it suffices to calculate a correlation score of the average observed correlation score. Otherwise you can choose between calculation of the correlation of a weighted average, of the weighted correlation of all sizes, the last approach being more costly and probably yielding a similar result.

When making use of the correlation scores, we see some strikingly results, that show that embryonic development is only taking place in the adult stage of the worms' life.

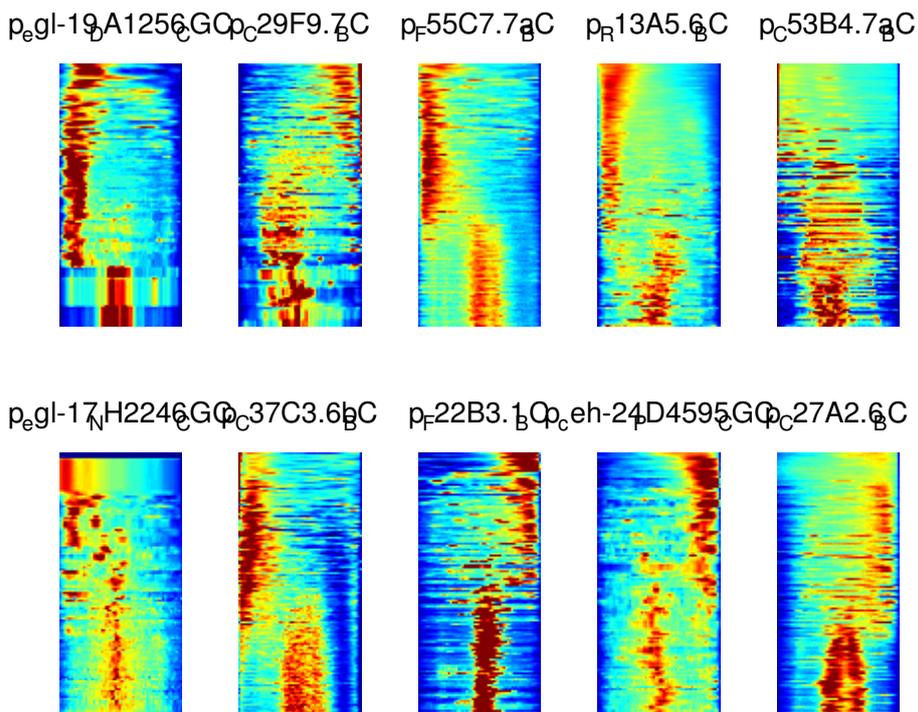


Figure 16: Embryonic development, found with correlation scores of mean expression.

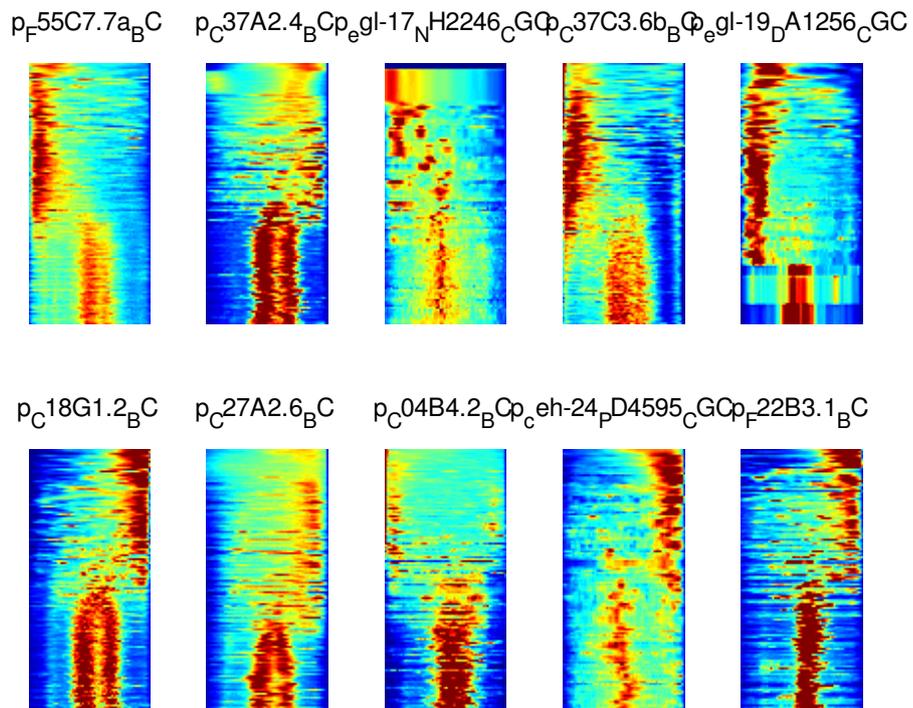


Figure 17: Embryonic development, found with mean correlation scores of individual expressions.

Mean of correlations vs. Correlation of Mean

It does make a difference whether you calculate correlation scores of individual worms and average those, or when you calculate the correlation score of the average expression. The first approach seems to find stronger enrichments.

Correlation works in two ways

While correlation shows good results when searching for enrichment, we can also make good statements on 'definitely *not* enriched'. As correlation scores that approach 1 are likely to be enriched, correlations that approach -1 are likely to be not enriched, but even are that opposite of that.

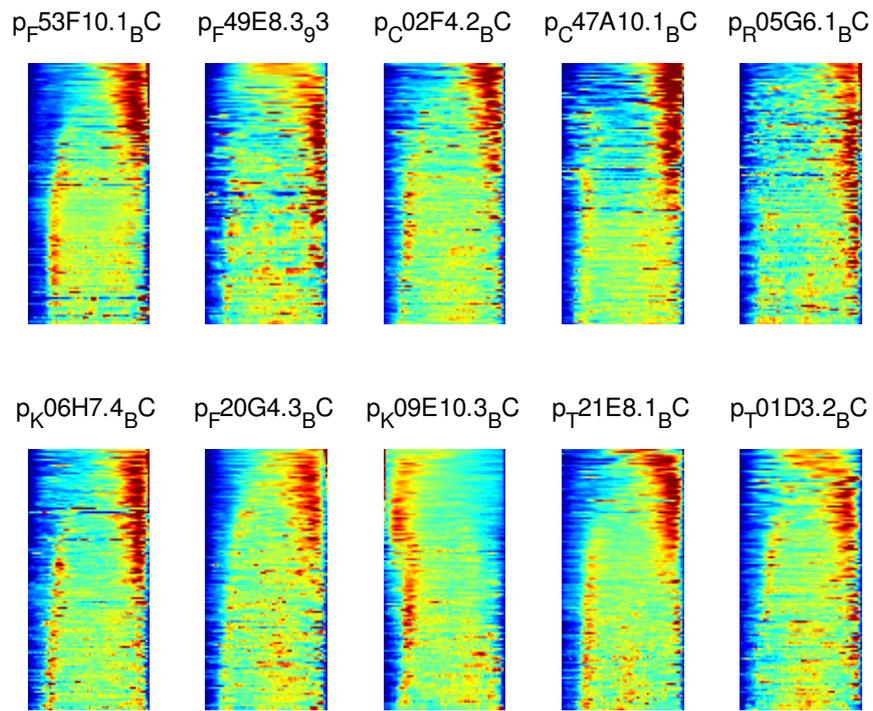


Figure 18: According to the correlation scores, following profiles were found to be not enriched in the head.

Signal Distributions of 'noise' vs. 'enrichment'. We set the signal area to 'head' because we believe that promoter p_C04C3.3_BC is enriched in the head.

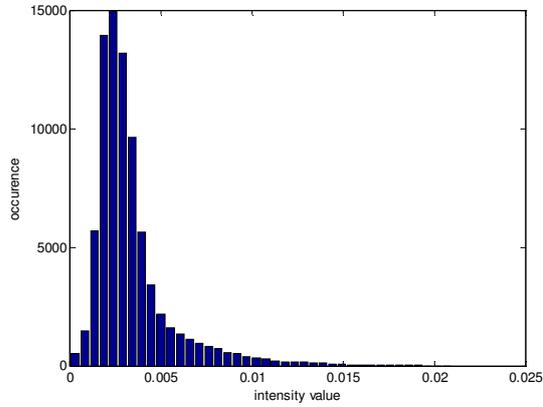


Figure 19: 'Noise' area distribution of promoter p_C04C3.3_BC

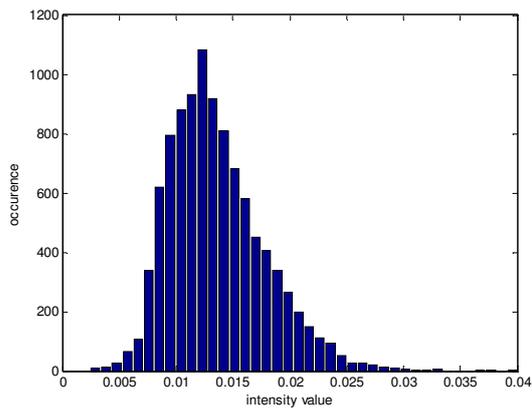


Figure 20: 'Signal' area distribution of promoter p_C04C3.3_BC

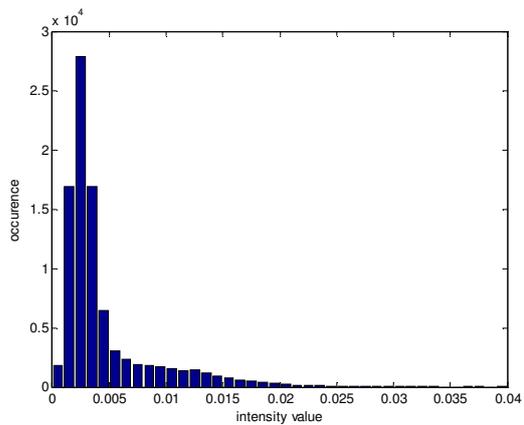


Figure 21: Complete distribution of promoter p_C04C3.3_BC

Signal Distributions of 'noise' vs. 'enrichment'. We set the signal area to 'tail', but we believe that promoter p_C04C3.3_BC is enriched in the head.

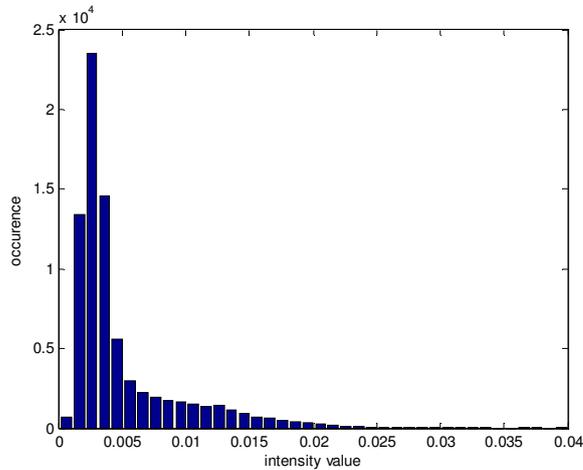


Figure 22: 'Noise' area distribution of promoter p_C04C3.3_BC

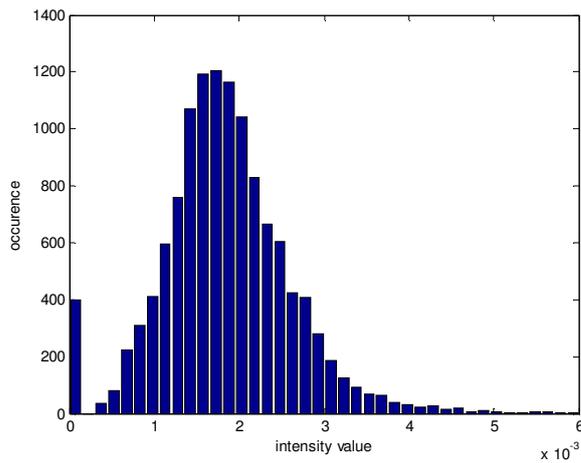


Figure 23: 'Signal' area distribution of promoter p_C04C3.3_BC

From Figure 19 to Figure 23 distributions, we cannot make any assumptions about what kind of distribution we see. The signal areas of both the 'right' area as the 'wrong' area seem to approach a normal distribution. The cumulative and 'noise' areas both are not normally distributed.

Can we test for enrichment with a t-test like test?

We need a definition for 'enrichment' first. (Cite Guoqiang Gu, Global expression analysis of gene regulatory pathways) For example. For enrichment we expect at least a threefold expression over background signal.

We can then make a null hypothesis that looks like $H_0 = \mu_E < 3\mu_B$

If that hypothesis can be rejected, we can safely predict that the observed mean expression in the area of interest is at least three times larger than the background signal: $H_1 = \mu_E \geq 3\mu_B$

Pros: You get signals that are at least threefold enriched.

Cons: Threefold is (very) arbitrary. See for instance top right histogram in Figure 11, where the expression profile is hardly more intense than the background signal. We still want to be able to say something on localized expression there, although probably not a very *intense* signal, it still can be a *specific* one.

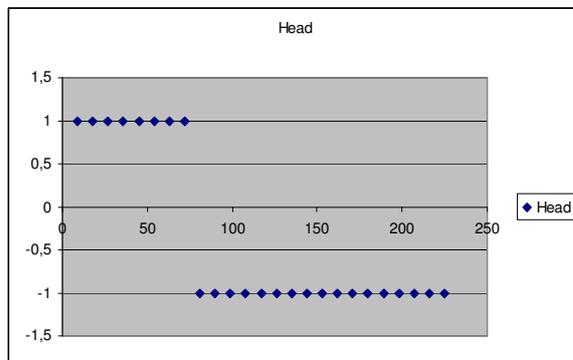
Correlation vs t-Test

The way in which we designed our correlation, will yield results that are in relation with the t-Test

$$\mu_x = E(x)$$

$$C(i, j) = E((i - \mu_i)(j - \mu_j)) = E(ij) - E(i)E(j)$$

When having $j = \begin{cases} 1, j_{inorgan} \\ -1, j_{outorgan} \end{cases}$ and $X_1 = i_{inorgan}, X_2 = i_{outorgan}$



$C(i,j)$ will transform to

$$C(i, j) = \bar{X}_1 - \bar{X}_2 - \mu_i \mu_j$$

$$\rho(i, j) = \frac{C(i, j)}{\sqrt{C(i, i)C(j, j)}}$$

The correlation is then related to the t-Test and will give similar results.

$$t = \frac{\bar{X}_1 - \bar{X}_2}{S_{X_1 X_2} \cdot \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

$$S_{X_1 X_2} = \sqrt{\frac{(n_1 - 1)S_{X_1}^2 + (n_2 - 1)S_{X_2}^2}{n_1 + n_2 - 2}}$$

