Shape Analysis of Brain Structures

Nicolas Tiaki Otsu

Kongens Lyngby 2011
IMM-B.Sc.-2011-04

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk

Summary

This bachelor thesis sets out to look into and analyze part of an extensive collection of data from the LADIS (Leukoaraiosis And DISability) Study. This data collection contains 1) bitmap images of mid-sagittal magnetic resonance images of human brains, 2) associated, expert-reviewed landmarks signifying the contour of the brain structure corpus callosum, 3) clinically assessed parameters evolved from tests done to the scanned persons. The analysis focuses on a) performing a sparse principal component analysis (SPCA) on the landmarks to describe local atrophical changes in the corpus callosum contour outline over a period of three years and also, on b) performing a regression analysis between these described local shape changes and the clinical parameter changes during the same period.

The analysis is carried out in MATLAB and leads to results that point towards connections between clinical parameters describing gait speed, executive motor control, verbal fluency and geriatric depression scale. The overall results show fairly acceptable similarities with those described in literature of research groups who performed both similar and non-similar analyses for describing correspondende between corpus callosum changes over time in correlation with clinical observations.

Resumé

Denne bachelorafhandling har til hovedformål at undersøge og analysere dele af en omfattende kollektion af data fra LADIS-studiet (Leukoaraiosis And DISability). Datamaterialesamlingen indeholder 1) bitmap-billeder af mid-sagittale magnetisk resonans-skanninger af menneskehjerner, 2) dertil knyttede, ekspertreviderede landmærker der betegner konturen af hjernestrukturen corpus callosum (hjernebjælken), 3) klinisk vurderede parametre udviklet fra forsøg udført på de samme skannede personer. Analyserne fokuserer på a) at udføre en sparsom principalkomponentanalyse (SPCA) på landmærkerne for at beskrive lokale, atrofiske ændringer i hjernebjælkekonturer over en tidsperiode på tre år, samt, b) på at foretage en regressionsanalyse mellem disse beskrevne, lokale formændringer og de kliniske parameterændringer i den samme periode.

Analysen er gennemført i MATLAB og fører til resultater der peger på sammenhænge mellem kliniske parametre beskrivende ganghastighed, udøvende motorkontrol, talefærdighed, samt geriatrisk depressionsskala. Det samlede resultat viser nogenlunde acceptable ligheder med dem beskrevet i litteratur af forskningsgrupper der har udført både lignende og anderledes analyser til beskrivelse af sammenfald mellem hjernebjælkeændringer over tid i korrelation med kliniske observationer.

Preface

This thesis was prepared at the section for Image Analysis, Informatics and Mathematical Modelling, the Technical University of Denmark as a partial fulfillment of the requirements for acquiring the degree Bachelor of Science in Engineering (B.Sc.Eng). The work amounts to 15 ECTS points and was carried out over a period of 5 months.

Lyngby, June 2011

Nicolas Tiaki Otsu

Acknowledgements

I thank my academic supervisor, professor in image analysis at DTU, Rasmus Larsen for his inspiration and invaluable help throughout the entire project. A warm thank you is also deserved by and given to professor in image analysis at DTU, Knut Conradsen for his guidance and support through the Friday meetings in the image analysis group.

Also, I would like to thank associate professor in image analysis at DTU, Rasmus R. Paulsen for his always friendly and focused encouragement towards training presentation techniques.

I also thank two specific Ph.D. students at DTU and the Danish Research Centre for Magnetic Resonance (DRCMR) located at Hvidovre Hospital in Copenhagen. Namely, Betina Vase Jensen for providing me with the data for the shape analysis, and Arnold Skimminge for providing me with a historical overview of the provided data and for advising me to narrow down the analytical aims for the project.

I thank external lecturer Karl Sjöstrand for making his *sparse principal component analysis* MATLAB toolbox publically available.

Lastly, I thank my partner, Astrid, for her endless love and patience with me when burdens seemed too heavy and for showing me that they weren't.

Contents

Summary i

Resumé iii

Preface v

Acknowledgements vii

1 Introduction 1
- 1.1 Background . . . 3
- 1.2 Motivation . . . 4
- 1.3 Thesis structure . . . 5

2 Theory 7
- 2.1 Landmarks distances . . . 7
- 2.2 Principal component analysis . . . 8
- 2.3 Sparse principal component analysis . . . 9
- 2.4 General linear model . . . 12

3 Implementation 15
- 3.1 Description of the data . . . 15
- 3.2 Description of the available Matlab packages and functions that are used . . . 18
- 3.3 Data analysis carried out in Matlab . . . 20

4 Results and Evaluation 29
- 4.1 Results . . . 29
- 4.2 Evaluation . . . 30

5 Discussion 37

6 Conclusion 39

A Additional Figures 41

B Matlab Code Listings 45

C Beta values and p values 63

Chapter 1

Introduction

In the modern society we may expect a longer life expectancy than in previous generations. This implies that more people will live longer and in consequence, the impact on society from neurodegenerative diseases such as Alzheimer's or dementia will increase. Besides the efforts put in by researchers and physicians towards treatment, there is also a significant need of methods and tools for diagnosing whether a person is in danger of developing such a disease.

The LADIS study is based on a collaboration between 11 European hospitals and consists of clinical tests and neuropsychological assessments of over 600 male and female individuals aged 65 to 84 evaluated with three year intervals. Together with the assessments, mid-sagittal brain MRi and CT scans have been made.

One of the purposes behind the study is

> *To evaluate age-related cerebral white matter changes (ARWMC) as independent determinant of the transition from healthy status to disability in elderly individuals.*

In the human brain, the largest collection of transversal nerve fibers are found in a white matter structure, called the *Corpus callosum* (CC). It is evident that the different parts of the CC contains nerve fibers which conduct highly

specific information. Figure 1.1 shows a human mid-sagittal magnetic resonance imaging (MRi) slice with 5 CC subdivisions. The topographic parts are named ([3]):

- **CC1** rostrum and genu,
- **CC2** rostral body,
- **CC3** midbody,
- **CC4** isthmus and
- **CC1** splenium.

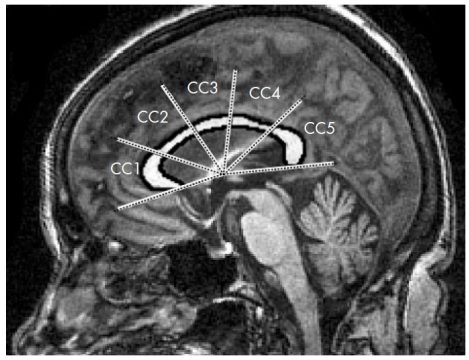

Figure 1.1: Example of mid-sagittal magnetic resonance imaging slice with 5 subdivisions of corpus callosum. Image originates from [2].

The aim of this thesis is to determine correspondences between CC shape changes and clinical data collected in the LADIS study.

The present work not only lays the foundation for the bachelor thesis of the author, but will hopefully also contribute to the LADIS work and provide new insight into the ways of determining the correlation between the corpus callosum shape changes due to atrophy and the corresponding clinical data. The 11 hospitals are:

- Helsinki, Finland (Memory Research Unit, Department of Clinical Neurosciences, Helsinki University)
- Graz, Austria (Department of Neurology and Department of Radiology, Division of Neuroradiology, Medical University Graz)

- Lisboa, Portugal (Serviço de Neurologia, Centro de Estudos Egas Moniz, Hospital de Santa Maria)
- Amsterdam, The Netherlands (Department of Radiology and Neurology, VU Medical Center)
- Goteborg, Sweden (Institute of Clinical Neuroscience, Goteborg University)
- Huddinge, Sweden (Karolinska Institutet, Department of Neurbiology, Care Sciences and Society; Karolinska University Hospital Huddinge)
- Paris, France (Department of Neurology, Hopital Lariboisiere)
- Mannheim, Germany (Department of Neurology, University of Heidelberg, Klinikum Mannheim)
- Copenhagen, Denmark (Memory Disorders Research Group, Department of Neurology, Rigshospitalet, and the Danish Research Center for Magnetic Resonance, Hvidovre Hospital, Copenhagen University Hospitals)
- Newcastle-upon-Tyne, UK (Institute for Ageing and Health, Newcastle University)
- Florence, Italy (Coordinating centre, Department of Neurological and Psychiatric Sciences, University of Florence)

1.1 Background

The human brain has many white and gray matter clusters and tracts that serve various purposes. Interconnecting nerve fibers each contribute an incomprehensible variety of functional and cognitive manifestations. The basic nerve signals are motoric, sensory and autonomous, and in combination they give rise to muscular contraction, both voluntary and reflectory, conscious unconscious senses. Explanations to phenomena such as emotions are not easily understood.

In [3], Ryberg et al. looked for significant correlation between local CC area changes and assessment on certain clinical parameters, including subjective memory complaints, geriatric depression scale (GDS) score and walking speed.

Due to the organization of the nerve fibers in CC, certain clinical observations should be expected to correlate with white matter hyperintensities (WMH) in specific parts of the CC in the median midsagittal plane. Jokinen et al. [2] describe age-related WMH as nerve cell atrophy as seen on magnetic resonance

images. Atrophy is the concept of nerve cell axons deteriorating and loosing their conductivity ability. Several million nerve fibers are contained within the median transversal tract called corpus callosum

Jokinen et al. [2] has described how The Danish Research Center for Magnetic Resonance at Hvidovre Hospital in Copenhagen has used a learning-based active appearance model that has been able to automatically locate and segment the mid-sagittal corpus callosum contour. The model contour is then described by landmark coordinates. Subsequently, an expert has adjusted these landmarks for inaccuracies. Figure 1.2 shows an example of how well these landmarks describe the corpus callosum contour outline.

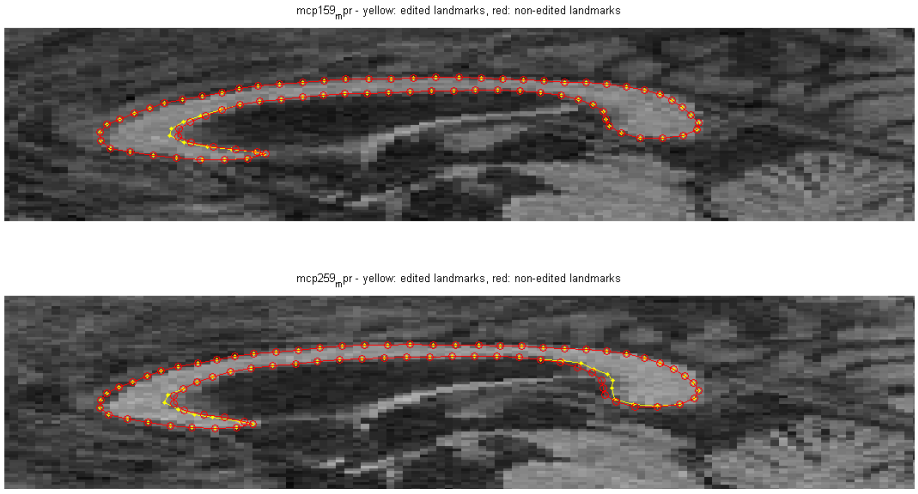

Figure 1.2: Top: close-up of CC baseline MR scan of test person CP59. Red shape represents landmarks computed by the learning-based active appearance model and yellow shape represents the expert corrected landmarks. Bottom: same as top for the follow-up scan. This test person was not sorted away in reduction step 3 described in Chapter 3.3.1.

1.2 Motivation

The Magnetic Resonance Images and associated clinical assessments emerged from the LADIS study provides excellent material for data analysis. In particular, the nerve fiber type organisation of the corpus callosum makes it meaningful to expect that clinical symptoms may manifest themselves as local white matter hyperintensities.

Instead of subdividing corpus callosum and looking into the volume changes, a method of subspace projection of the landmark coordinate changes named *sparse principal component analysis* may lead to local shape change representations that are directly comparable to changes in clinical neuropsychological performance parameters.

Based on the idea that local shape changes may correspond with certain clinical manifestations, this method is worth looking into and trying to understand. One of the advantages of sparse principal component analysis is that one can enforce the variation in corpus callosum shape change to be explained by a selected number of landmark coordinate changes.

The methods for selecting, or regressing, the number of variables, the landmark coordinate changes in this case, have been refined over several years by different authors, which will be described in the Theory Chapter. An analysis on the ways of performing sparse principal component analysis by regressing on the variables derived from a regular principal component analysis and performing a regression analysis of this outcome and the clinical observations will be the overall purpose of this thesis.

1.3 Thesis structure

The present chapter has described the background for the interest in analyzing the shape changes of the corpus callosum in a mid-sagittal view. It has also stated the motivation that drives the author towards working on method for analyzing these shape changes.

Chapter 2 describes the theoretical background for the analytical work of the thesis.

Chapter 3 describes how the theory has been implemented.

In Chapter 4, the results from the analysis are presented and evaluated.

The theoretical parts of the thesis involves the use of certain mathematical entries that will be described below:

- Bold lower-case entries describe vectors. Example: $\mathbf{b}$.
- Bold upper-case entries describe matrices. Example: $\mathbf{Z}$.

- Subscripts denote dimensionality. Vector example: $\mathbf{b}_i$. Matrix example: $\mathbf{Z}_{i,j}$.
- Italized lower entries describe scalars. Example: s.
- Greek letters (with dimension subscript) denote random model coefficients. Example: β_i.
- The number of observations is denoted by n.
- The number of variables is denoted by p, or if number is altered, k.

Chapter 2

Theory

This section describes the theory that lays the foundation for implementation of the work carried out in the thesis work.

2.1 Landmarks distances

When working with mid-sagittal magnetic resonance images which are two-dimensional, it is import to notice that the landmarks describing the contour of interest may not necessarily be of the same scale and position in the images. This needs to be corrected for, and in this thesis, the methods for aligning these structures involve two steps: *centering* and *normalization*, as adapted from [4].

Centering: By collecting a set of landmarks in a column vector $\mathbf{x}$ we can write it: $\underset{n\times p}{\mathbf{x}_{ij}} = [\mathbf{x}_{x1}, \mathbf{x}_{x2}, \cdots, \mathbf{x}_{x78}, \mathbf{x}_{y1}, \mathbf{x}_{y2}, \cdots, \mathbf{x}_{y156}]^T \in \mathbb{R}^{n\times p}$ with n being the number of observations, p be the number of variables (156 landmark coordinates) and the indices x, y being the first- and second coordinate. Letting $\underset{1\times 1}{\overline{\mathbf{x}}_j} = \frac{1}{n}\sum_{i=1}^{n} \underset{n\times p}{\mathbf{x}_{ij}}$ enables us to compute the centered $n \times 1$ column vectors

$\mathbf{x}_{\text{cent}j}$ by Equation 2.1

$$\underset{n\times 1}{\mathbf{x}_{\text{cent}j}} = \underset{n\times 1}{\mathbf{x}_j} - \underset{n\times 1}{k} \cdot \underset{1\times 1}{\overline{\mathbf{x}}_j}\,, \tag{2.1}$$

with k being the unit $n \times 1$ one-vector: $k = \mathbf{1} \in \mathbb{R}^{n\times 1}$.

Normalization is performed after centering the landmark matrix $\mathbf{x}_{\text{cent}ij}$ to ensure that the sum of all columns become of unit length 1. Equation 2.2 shows the calculation:

$$\underset{n\times 1}{\mathbf{x}_{\text{norm}j}} = \frac{1}{\sqrt{\underset{n\times 1}{k} \cdot \sum_{i=1}^{n} \underset{1\times 1}{\mathbf{x}^2_{\text{cent}ij}}}} \underset{n\times 1}{\mathbf{x}_{\text{cent}j}}, \tag{2.2}$$

such that $\sqrt{\sum_{i=1}^{n} \underset{1\times 1}{\mathbf{x}_{\text{norm}ij}}} = 1$. This can be interpreted such that the sum of all squared column elements equals 1. Figures 3.2 and 3.3 show the landmark normalization procedure.

2.2 Principal component analysis

Here follows a description of the theory of *principal components analysis* as adapted from Sjöstrand et al. [5], [6], Zou et al. [8] and Sjöstrand [4].

When the CC have been segmented and normalized (scaled and centered), the number of landmarks/variables p (see Figure 2.1) distributed among the CC outlines (corresponding to the number of images/observations n) can be collected in an $\underset{n\times p}{\mathbf{X}}$ data matrix. The variables are non-orthogonal and will span a p-dimensional hyperplane of unknown correlation (linear dependency). If we want to find out in which directions the CC outlines differ most from each other, the PCA is a method of rotating the data matrix such that the variance in each direction can effectively be identified.

The rotation of $\underset{n\times p}{\mathbf{X}}$ is done by use of a rotation matrix $\underset{p\times k}{\mathbf{B}}$ in which the columns are called loading vectors. The rotation results in a matrix $\underset{n\times k}{\mathbf{Z}}$ in which the

columns are the principal components (PCs), as seen in Equation 2.3.

$$\underset{p\times k}{\mathbf{Z}} = \underset{n\times p}{\mathbf{X}}\,\underset{p\times k}{\mathbf{B}} \tag{2.3}$$

The number $k \leq p$ signifies the number of loading vectors that are utilized in the rotation. Together, the loading vectors describe p orthogonal directions along which the variations in the data set are distributed. The total variation for the entire data set is described by the sum of variation for all principal components. By sorting these in descending order (and making the same sorting of the rotation matrix $\mathbf{B}$) and summing from the highest variation towards the lowest, the most significant desired variation percentage can be obtained. Spanning vectors by linear combinations of the principal components result in completely linearly independent vectors which can lead to a better perception of the differences between the CC outlines.

The principal component matrix can be computed by computing the covariance matrix of $\mathbf{X}$ and performing an eigenanalysis on this matrix. It can also be computed by performing a singular value decomposition (SVD) of $\mathbf{X}$ such that

$$\underset{n\times p}{\mathbf{X}} = \underset{n\times n}{\mathbf{U}}\,\underset{n\times p}{\mathbf{D}}\,\underset{p\times p}{\mathbf{V}^T}. \tag{2.4}$$

Performing a SVD on $\mathbf{B}$ as described in Equation 2.4, the PCs $\mathbf{Z}$ is the product of $\mathbf{UD}$. The $p \times p$ matrix $\mathbf{V}$ contain the loadings that correspond to the PCs.

When distributing landmarks along the CC outlines, some of these landmarks will correspond better than others when comparing images, see Figure 2.1. Since PCA takes every single variable (landmarks coordinates, in this case) into consideration, some of the variation that the PCs take into consideration may not be easily interpretable in term of CC shape variation and are therefore difficult to use for shape analysis. This gives rise to a popular interest in investigating methods for determining which principal components to use in the analysis. The focus of this thesis is to investigate the concept of *sparse principal components analysis*.

2.3 Sparse principal component analysis

The following theory of *sparse principal components analysis* is adapted from Sjöstrand et al. [5], [6], Zou et al. [8] and Sjöstrand [4].

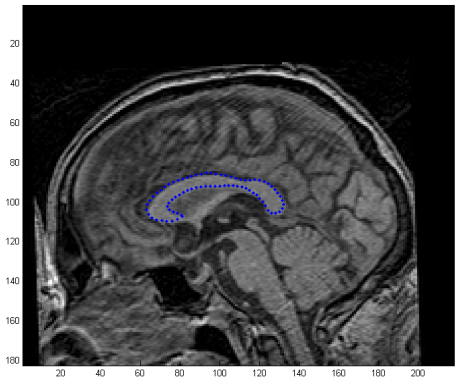

Figure 2.1: Sample of 78 landmarks along a corpus callosum outline.

When investigating local shape changes with respect to landmark coordinates, it is crucial to use a method of applying sparsity to the landmark perturbations performed by the loadings associated with regular principal components. Based on the fact that each principal component computed by a regular principal component analysis are in fact correlated with the landmark coordinates and weighed by the loadings, sparsity can be enforced by regressing on these principal components.

The *elastic net regression* in Equation 2.5 is a method of forcing the right- and lefthand side elements of Equation 2.3 towards zero in a manner that the number of non-zero elements is controlled:

$$\mathbf{b}_i = \arg\min_{\mathbf{b}_i} \|\mathbf{z}_i - \mathbf{X}\mathbf{b}_i\|^2 + \lambda \|\mathbf{b}_i\|^2 + \delta \|\mathbf{b}_i\|_1 , \tag{2.5}$$

in which $\|\cdot\| = \|\cdot\|_2 = \sqrt{\sum_{i=1}^{p} \cdot_i^2}$ signifies the squared 2-norm Euclidean distance, ℓ_2, and $\|\cdot\|_1 = \sum_{i=1}^{p} |\cdot_i|$, the 1-norm, ℓ_1 of $\cdot$.

In Equation 2.5, $\|\mathbf{z}_i - \mathbf{X}\mathbf{b}_i\|^2$ describes the residual distance between the i'th principal component $\mathbf{x}_i$ of $\mathbf{Z}$, called the response variable, and the perturbation on $\mathbf{X}$ by the i'th loading vector, $\mathbf{x}_i$, denoted as the predictor variable. If $\lambda = 0$, we are left with the so-called *LASSO* method, which stands for *Least Absolute Shrinkage and Selection Operator method*. When the number of variables, p is higher than the number of observations, n, the *LASSO* method is able to force each of the coefficients in $\mathbf{b}$, to zero as δ grows larger. As as more coefficients

are turned to zero, a desired number of remaining non-zero coefficients can be detected. If $\delta = 0$, we have the so-called *ridge* regression, which is a method of shrinking the coefficients of $\mathbf{b}$.

The elastic net regression computes a sparse loading vector $\mathbf{b}$ that is close to the response variable, $\mathbf{z}$. These loading vectors suffer from the lack of the property that the regular, non-sparse loading vectors have: they are at right angles with each other, which is why they are able to describe variance in each their own principal direction. To make the sparse loadings assimilate the orthogonal properties of the non-sparse loadings, Zou et al. [8] have formulated a "*SPCA Criterion*", formulated in Equation 2.6:

$$\begin{aligned}(\widehat{\mathbf{A}}, \widehat{\mathbf{B}}) =& \arg\min_{\mathbf{A},\mathbf{B}} \sum_{i=1}^{n} \left\| \mathbf{x}_i - \mathbf{A}\mathbf{B}^T\mathbf{x}_i \right\|^2 + \lambda \sum_{j=1}^{K} \|\mathbf{b}_j\|^2 + \sum_{j=1}^{K} \delta_{1,j} \|\mathbf{b}_j\|_1 \\ & \text{subject to } \mathbf{A}^T\mathbf{A} = \mathbf{I}_{K\times K}.\end{aligned} \tag{2.6}$$

The implementation of this criterion is done by setting $\mathbf{A}$ equal to the first K loadings of the ordinary principal components. All sections of Chapter 3 refer to exactly this number, K. By denoting $\mathbf{A}$ as $\mathbf{A} = [\alpha_1, \cdots \alpha_K]$ and solve the so-called elastic net problem in Equation 2.7 (first iteration step):

$$\mathbf{b}_j = \arg\min_{\mathbf{b}} (\alpha_j - \mathbf{b})^T \mathbf{X}^T\mathbf{X}(\alpha_j - \mathbf{b}) + \lambda \|\mathbf{b}\|^2 + \delta_{1,j} \|\mathbf{b}\|_1 \tag{2.7}$$

for $j = 1, 2, \cdots, K$, by fixing $\mathbf{B}$ to $\mathbf{B} = [\mathbf{b}_1, \cdots \mathbf{b}_K]$, a singular value decomposition can be computed of $\mathbf{X}^T\mathbf{X}\mathbf{B} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ as described in Equation 2.4. Then update $\mathbf{A} = \mathbf{U}\mathbf{V}^T$ (second iteration step). By performing first and second iteration steps for a preset λ value, the method makes it possible to iterate until a desired number of non-zero δ coefficient are found. These coefficients decide what $\mathbf{b}$-coefficients are kept non-zero in the loading vectors from the regular principal component analysis. Computing non-zero $\mathbf{b}$-coefficients in this way is the essence of making the principal component analysis sparse.

2.3.1 Deformation modes

When visualizing the effect a principal component analysis (sparse or non-sparse) has on a set of landmark coordinates, *deformation modes* will show

how the landmarks are perturbed.

$$\mathbf{x}_{\text{mode}i} = \overline{\mathbf{x}}_i + s \cdot \sqrt{\lambda_i} \cdot \mathbf{b}_i \tag{2.8}$$

Equation 2.8 shows the computation of the perturbed landmarks, also referred to as deformation modes, or *modes of variation*. $\overline{\mathbf{x}}_i$ is the i'th mean shape as described in Section 2.1, and s is an integer signifying the number of standard deviation perturbations. λ_i is the i'th eigenvalue and $\mathbf{x}_i$ is the i'th loading vector (i'th column in $\mathbf{B}$ in Equation 2.3).

2.4 General linear model

In order to compare the scores from the sparse principal component analysis with the clinical observations, a series of univariate test are performed. The target is to determine if there is a correspondence between the changes over time of the clinical observations and the computed sparse loadings. First, the response variable of the regression analysis is defined by

$$\underset{n\times var}{\Delta \mathbf{y}} = \underset{n\times 1}{\mathbf{PC}_{k,stop}} \cdot \underset{1\times 1}{\beta}\ , \tag{2.9}$$

for which n is the number of observations (test persons), var is the centered differences between the baseline and follow-up clinical variables which will be described further in Section 3.3.1. $k = 1, \cdots,$ K, where K is the number of computed principal components computed as described in Section 2.3. *stop* is integer values ranging from 2 to p which declare the number of desired non-zero components computed by the LASSO method. p is the number of variables (landmark coordinates).

The scores, PC in Equation 2.9 are the principal components and are computed by Equation 2.10

$$\underset{n\times 1}{\mathbf{x}_{k,stop}} = \underset{n\times p}{\Delta \mathbf{x}_{\text{norm}}} \cdot \underset{p\times 1}{\mathbf{b}_{k,stop}}, \tag{2.10}$$

for $k = 1$, ..., K, stop $= 2$, ... p and $\Delta\mathbf{x}_{\text{norm}} = \mathbf{x}(\text{baseline})_{\text{norm}} - \mathbf{x}(\text{follow-up})_{\text{norm}}$ for which the right-hand side is computed by Equation 2.2.

The full regression analysis will cover K = 10 sparse principal components, 20 stop values and 5 clinical difference variables, giving rise to $10 \times 20 \times 5$ $\beta = 1000$ values with 1000 corresponding p-values. These p-values each show the probability that there is a significance that a given score has the same mean as a given clinical difference variable.

Each p-value will be investigated for significance levels of $10, 5, 1$ and 0.1%.

Chapter 3

Implementation

The data analysis of the thesis has been implemented in Matlab. The present chapter is divided into three sections. The first section contains a description of the data provided by the Danish Research Centre for Magnetic Resonance (DRCMR). The second section describes the built-in Matlab functions that have been crucial to the computations. The section also contains a description of Karl Sjöstrand's publically available sparse principal component analysis software package. The third section describes the Matlab scripts and functions created by the thesis author for carrying out the data analysis.

3.1 Description of the data

The data used in the analysis covers three different file types:

bitmap Contains the actual image data,

mat contains the landmark coordinates, and

Excel contains the clinical assessment data.

Each type will be described more thoroughly in the following subsections.

3.1.1 The bitmap image files

The locations of the `bitmap` files are divided into three folders named `ccam`, `cccp` and `ccladis`. They contain image data for the hospitals in Amsterdam, Copenhagen and the remaining 9 hospitals as mentioned in Chapter 1. The actual magnetic resonance images are 8 bit grayscale of dimension 218×182. There are 978 images, summing to 489 test persons.

The naming convention for the images is shown in Table 3.1 and constitutes a form that contains three elements that are crucial to the present work: the hospital name code, the test person number of the hospital and an index telling whether the image arises from a baseline or a follow-up scan. These alternatives of these index codes are shown in Table 3.2.

Bitmap file name:	`mam101_mpr.bmp`
Index:	`-HhSTp----.bmp`

Table 3.1: Table shows an example of the naming of the **bitmap** images. The indeces **Hh**, **S** and **Tp** refer to the hospital name, time of scan and test person number, respectively. The file name points to the baseline scan of test person number 1 at the Amsterdam hospital.

Index term	Alternatives	Explanation
Hospital (Hh)	am	Amsterdam
	cp	Copenhagen
	fl	Florence
	gr	Graz
	gt	Gothenburg
	he	Helsinki
	hu	Huddinge
	ls	Lisboa
	ma	Mannheim
	nc	Newcastle-upon-Tyne
	pa	Paris
Time of scan (S)	1	baseline
	2	follow-up
Test person number (Tp)	01	Varies
	⋮	

Table 3.2: Table shows the different alternatives for the index terms of the naming of the **bitmap** image files shown in Table 3.1.

3.1.2 The mat files

The mat files are found within the same folders as the bitmap images and follow a similar naming pattern as that of the bitmap files, only with a different ending, as shown in Table 3.3. The index terms are the same as mentioned in Tables 3.1 and 3.2.

bitmap file name:	`mam101_mpr.bmp`
Corresponding mat file name:	`mam101_mpr_result`

Table 3.3: Table shows the correspondence in naming pattern for the bitmap image files and the mat files.

The mat files, when loaded into the MATLAB workspace, leads to one `char` type variable named `basename` which matches the bitmap file name shown in Table 3.3 without the `.bmp` ending. It also leads to 6 `double` type variables, out of which only the two variables named `landmarks` and `landmarks_edited` are used in the thesis work. These are both of size (78,2) and their columns hold the first and second axis landmark coordinates for the corpus callosum contour. The landmarks have been segmented by staff at the Danish Research Centre for Magnetic Resonance (DRCMR) at Hvidovre Hospital, using a learning-based active-appearance model which has subsequently edited by an expert to correct for the automatic segmentation ([5], [1], [7]).

3.1.3 The Excel files

All in all, there are 13 Excel datasheet files containing a vast amount of clinical assessments performed on 639 test persons. Out of these, the 14 assessments mentioned in Table 3.5 are used in the work in the present analysis.

The test person nomenclature of the Excel datasheets are different from that of the bitmap and mat files.

Excel database test person naming pattern:	`AM01`
Index:	`HhTp`

Table 3.4: Table shows the Excel datasheet naming pattern for the same test person as in Table 3.1. The indeces **Hh** and **Tp** refer to the hospital name and test person number, respectively. See Table 3.2 for explanation on the indeces.

Each of the 13 Excel datasheets contain test person names in the first column and clinical parameters in the first row. Table 3.5 shows which Excel files are used for catching the selected data.

Excel datasheet name	Contained clinical variables
compound_measures_wp4.xls	MEMORY MEM3y SPEED SPEED3y EXECUTIVE EXEC3y
table2_baseline.xls	verbal gdstotal
table2_3y.xls	verbal3y gdstotal3y
table1_baseline.xls	sex birthday daterif
table1_baseline.xls	datefu3

Table 3.5: Table shows which Excel datasheet files contain the selected clinical variables.

3.2 Description of the available Matlab packages and functions that are used

This section contains information about the MATLAB packages and functions that have been utilized in the thesis work.

xlsread. Built-in MATLAB function. The call: `[NUMERIC,TXT,~]= XLSREAD(FILE)` is used and reads into the data specified in the Excel .xls file named FILE. Two outputs are extracted, namely `NUMERIC`, a cell type variable holding the numeric datasheet values and `TXT`, a cell type variable holding the text datasheed values. `~` signifies a non-utilized output.

ismember. Built-in MATLAB function. The call: `ISMEMBER(A,S)` returns 1 where the elements of `A` are contained within the set `S` and 0 in the opposite case. The output is an array which has the same size as `A`. This function is used for detecting which columns from the imported Excel datasheets holds

info about which clinical variables.

findstr. Built-in MATLAB function. The call `FINDSTR(S1,S2)` finds the shortest of the two strings `S1` and `S2` and returns the starting indices in case the shortest string is contained within the longest. The function is used as a logical operator to decide if a string is contained within another in order to compare the test person names due to the different nomenclature occuring in the bitmap/mat files and Excel datasheets.

regstats. Function which is a part of MATLAB's `statistical toolbox`. The call: `STATS = REGSTATS(RESPONSES,DATA,MODEL,WHICHSTATS)` is used to carry out the regression analysis between the clinical variables and sparse principal scores. The `RESPONSES` input is the clinical variable vector $\Delta\mathbf{y}$ in Equation 2.9, and the `DATA` input is a principal component $\mathbf{PC}$ of same size as $\Delta\mathbf{y}$. The input `MODEL` is set to '`linear`' to enable the general linear model functionality. The input `WHICHSTATS` receives the cell array `{'fstat', 'beta'}` in order to catch the F-statistic p- and β values.

center. Function which is a part of Karl Sjöstrand's `sparse principal component analysis toolbox`. The call `X = CENTER(X)` computes and outputs the centered matrix of same size as the input, which is (n,p) with n being the number of observations (sets of landmarks) and p being the number of variables (landmark coordinates within each set). This function implements the calculations described in Equation 2.1 in Chapter 2.

normalize. Function which is a part of Karl Sjöstrand's `sparse principal component analysis toolbox`. The function is called by `X = NORMALIZE(X)`. The input data matrix is centered by utilizing the function **center** and scaled such that the columns have unit length. This function implements the calculations described in Equation 2.2 in Chapter 2.

svd. Built-in MATLAB function. The call: `[U,S,V] = SVD(X,'econ')` computes the singular value decomposition of the data matrix $\mathbf{X}$ from Equation 2.4 of the dimensions (n,p) in which n is the number of observations (landmark sets) and p, the number of variables (landmark coordinates). The input `'econ'` assures that in the case with the data used in the thesis work where $n>p$, only the p columns of $\mathbf{U}$ in the mentioned equation are computed. This also implies that the output, `S` ($\mathbf{D}$ in the equation), becomes of size (p,p).

larsen. Function which is a part of Karl Sjöstrand's `sparse principal component analysis toolbox`. The call: `BETA = LARSEN(X, Y, LAMBDA2, STOP, TRACE)` has the following inputs: `X` is the normalized (n,p) data vector where each column contains the set of landmark coordinates organized as mentioned in Chapter 2.1. The input response vector `Y` is the centered scores

contained in the **Z** matrix in Equation 2.3. The input `lambda` is the ridge regression coefficient described in Chapter 2.3. The input `stop` contains negative numerical integer values ranging from -2 to -156, corresponding to the desired number of nonzero variables (landmark coordinates) in the *LASSO* part of the elastic net regression framework. The input `trace` is set to zero and is not utilized in the present thesis work. The output `beta` contains the remaining, non-zero loading coefficients emerged from the elastic net regression.

spca. Function which is a part of Karl Sjöstrand's `sparse principal component analysis toolbox`. Main function for computing the sparse principal principal components and sparse loadings. The call: `[SL SV PCAL PCAV PATHS] = SPCA(X, Gram, K, LAMBDA, STOP)` has the following inputs: `X` is the (n,p) matrix with n observations (sets of landmark coordinates) and p variables (landmark coordinates). `Gram` is not utilized in the present thesis work. `K` is the desired number of principal components. The inputs `lambda` and `stop` are passed onto the function **larsen**. The outputs `PCAL` and `PCAV` are the regular principal component loadings (the columns of **B** in Equation 2.3 and corresponding principal components (the columns of **Z** in the same equation). The outputs `sl` and `sv` contain the sparse principal component loadings and corresponding sparse principal components whose number of non-zero elements are determined by the current `stop` number. The output `paths` is not utilized in the present thesis work.

3.3 Data analysis carried out in Matlab

This section describes how MATLAB has been used to perform the data analysis. The overall main.m MATLAB script which provides the basis for the scripts and functions described in the following subsections, is shown in Listing B.1. Please note the following when reading the code listings throughout the thesis:

- The sign ¬ corresponds to the sign ∼ when viewed in MATLAB.
- The sign Δ corresponds to the entry `delta` when viewed in MATLAB.

3.3.1 Selection of clinical variables

The LADIS foundation contains a vast amount of clinically assessed parameters, out of which those mentioned in Table 3.5 have been selected. Listing 3.1 shows the creation of `base`, `dir_bmp` and `dir_mat` structs for use when importing data

into MATLAB. The full MATLAB code for setting the file directories is shown in Listing B.2.

```
base.am='E:\LADIS\ccam\';
dir_bmp.am=dir([base.am '*.bmp']);
dir_mat.am=dir([base.am '*.mat']);
```

Listing 3.1: Matlab code for creation of structs for use when loading data into workspace.

There are three reducing steps involved in selecting test person data prior to performing the actual sparse principal component analysis on their associated corpus callosum contour landmarks.

First reduction step. No. of test persons reduced from 639 to 385:

When using the above mentioned structs, the MATLAB script `clinimp.m` (shown in Listing B.3) imports the selected clinical variables and checks if all test persons have both baseline and follow-up assessments of the variables. The remaining test person data is stored in the structs `clin` and `full` and are saved in the mat file `clinimp.mat`.

`clin` contains 3 fields: `vars`, a cell with 14 fields: the first 10 are those described in Table 3.5, and the remaining 4 are `age`, `age3y`, `male` and `female`. The gender information comes from the column `sex` in `table1_baseline.xls` and `age` and `age3y` has been computed from `birthday`, `daterif` and `datefu3`. The MATLAB code for computation of the test person `age` is shown in Listing B.4. `clin` also contains a double field named `num` in which the rows correspond to the test persons and the columns correspond to the 14 variables in the `var` field. The last field in the `clin` struct is a cell with the test person names as they appear in the Excel datasheets.

`full` is a struct with 2 fields: `num` and `txt` which hold the full data extracted from the 5 Excel datasheets mentioned in Table 3.5.

Second reduction step. No. of test persons reduced from 385 to 216:

The function `convertname.m` in Listing B.5 utilizes the three functions `clinred.m`, `getcoords.m` and `deltaclin.m` shown in Listings B.6, B.7 and B.8 to do the following steps of reduction:

1. Convert the Excel datasheet test person names to that of the bitmap and mat files.

2. Find matching stringnames for the found test person names that imply that they have both baseline and follow-up MR scan performed on them.
3. Create `LMi` struct containing indices showing the which bmp and mat files that match with each other for the reduced list of test persons for both baseline and follow-up.
4. Reduce the number of rows in `clin.num` and `clin.name` to match the test person names that are left after the second reduction step.
5. Use the `LMi` index struct to get the learning-based active appearance model computed landmarks and also the expert editions of those landmarks. This step is performed by `getcoords.m` which generates a struct `x` with non-edited landmarks, edited landmarks and distance computations of those landmarks.
6. Compute the differences between the baseline and follow-up performance assessments for the found test persons.

Third reduction step. No. of test persons reduced from 216 to 205:

The final reduction step is prepared via the script `inspect.m` shown in Listing B.9. The script displays both baseline and follow-up bitmap images together with both the non-edited and edited landmarks. By using the built-in MATLAB function `waitforbuttonpress.m` which returns zero when a mouse button is pressed and 1 if a keyboard button is pressed, the script generates a logical list of test persons whose edited landmarks have been visually inspected and validated. The script then reduces the index struct `LMi` and also creates a vector `sortindex` for later use when doing the actual test person reduction in the `clin` struct. One of the test persons whose edited landmarks were accepted is shown in Figure 1.2. The images and edited landmarks for a person who was sorted away, is shown in Figure 3.1. Figures A.1, A.2, A.3 and A.4 in Appendix A shows some more visual inspection plots.

The actual reduction of the final step is performed by the function `clinred_02` in Listing B.10. This function takes the input `clin` and `sortindex` and outputs the final, reduced struct `clin`.

3.3.2 Preparing the observations for sparse principal component analysis

The index struct `LMi` used for extracting the final clinical variables is now used in association with the data struct to and the function `getcoords.m` to get

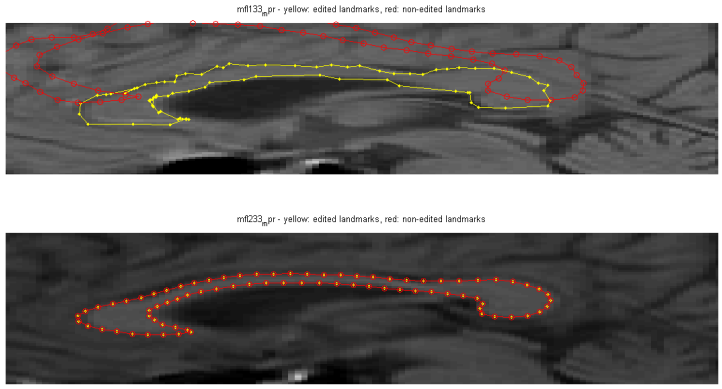

Figure 3.1: Top: close-up of CC baseline MR scan of test person FL33. Red shape represents landmarks computed by the learning-based active appearance model and yellow shape represents the expert corrected landmarks. Bottom: same as top for the follow-up scan. This test person was sorted away in reduction step 3 described in Chapter 3.3.1.

the final reduced landmark coordinates. Namely, the double type numerical matrix `x.ed.delta_norm` containing the edited landmark coordinate changes from baseline to follow-up, are of interest. Figure 3.2 shows the 205 edited landmark coordinates before normalization in red and after normalization in green. Figure 3.3 shows a close-up of the normalized coordinates.

The function `spca.m` is now iteratively called with the following input parameters:

- `x.ed.delta_norm`. The matrix is transposed such that it has the dimensions (n,p) and represents $\mathbf{X}$ in Equation 2.4.
- `trace = 1`. This ensures that the function prints information in the MATLAB command window.
- `K = 10`. This is the selected number of sparse principal components.
- `lambda = 1`. This is the ridge regression parameter from Equation 2.5.
- `stop = -round(linspace(2,156,20));` contains $j = 20$ integer values which make out the changing iteration parameters. Each iteration's integer value corresponds to the number of desired non-zero variables. This input varies the effect of the *LASSO* method in Equation 2.5.

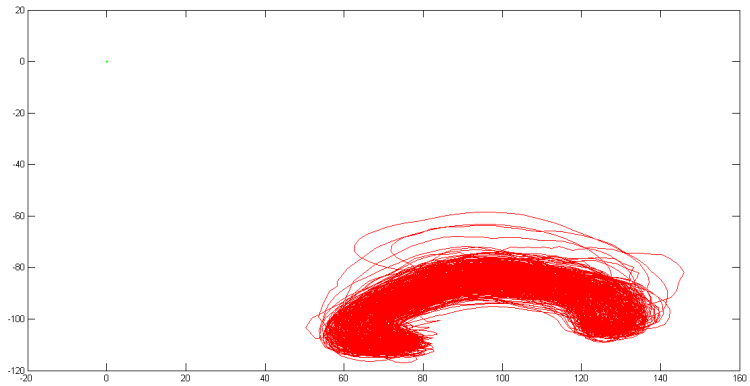

Figure 3.2: Red: Baseline edited landmarks for the 205 test persons. Green: Normalized versions of the same landmarks. The normalization procedure has centered the landmarks around (0,0) and scaled to assure unit length of invidual corpus callosum shapes.

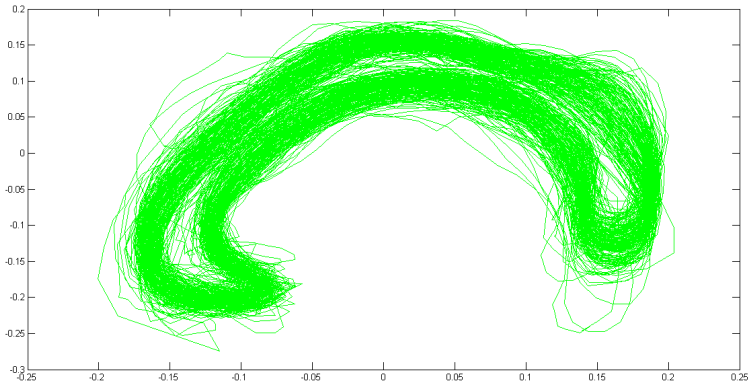

Figure 3.3: Zoom of green structures in Figure 3.2 showing normalized baseline edited landmarks for the 205 test persons.

The actual sparse principal component analysis procedure call in MATLAB is shown in Listing 3.2.

```
% Compute SPCA on the normalized, edited landmark coordinates
maxiter = 150;
trace   = 1;
lambda  = 1;
stop    = -round(linspace(2,156,20));
K       = 10;
for i = 1:length(stop)
    [a b c d ¬] = spca(x.ed.Δ_norm', [], K, lambda, stop(i), ...
        maxiter, trace);
    SPCA.sl.K10(i).norm = a;
    SPCA.sv.K10(i).norm = b;
    SPCA.pcal.K10.norm  = c;
    SPCA.pcav.K10.norm  = d;
end

% Reorganize structure of SPCA
for i = 1:length(stop)
    spca.sl(i).k10 = SPCA.sl.K10(i).norm;
    spca.sv(i).k10 = SPCA.sv.K10(i).norm;
end
spca.pcal = SPCA.pcal.K10.norm;
spca.pcav = SPCA.pcav.K10.norm;
```

Listing 3.2: Iterative computation of sparse and non-sparse principal components and loadings.

The output is collected in a struct, `SPCA`, with four fields:

sl contains $1 \times j = 20$ struct array `K10` in which each field is a $p \times K$ (156×10) double type field named `norm`. These 20 struct fields with each 10 contain 200 sparse loadings made up of 10 loadings with each 20 different non-zero variable numbers held in the `stop` input.

sv is built up in the same way as the field **sl**, only the double field norm has the dimension $K \times 1$ (10×1) sparse eigenvalues corresponding to the loadings in **sl**.

pcal contains the regular, non-sparse loadings corresponding to the regular loadings described in Equation 2.4 as the matrix $\mathbf{V}$ with the dimension $p \times p$ (156×156).

pcav contains the p regular, non-sparse eigenvalues corresponding to the loadings in **pcal**.

3.3.3 Performing regression analysis on the scores and clinical variables

In order to compute the scores as in Equation 2.10, a struct of size $K \times j = 10 \times 20 = 200$ in which K is the number of sparse principal components and j is the number of non-zero loading elements is computed. It is done iteratively, and the main part of the code is shown in Listing 3.3. In this way, each struct field has the size $(n \times 1) = (205 \times 1)$ as the mentioned equation prescribes.

```
for i = 1:r
    for j = 1:c
        score{i,j} = spca.sl(j).k10(:,i)'*x.ed.Δ_norm;
    end
end
```

Listing 3.3: Main part of code for computing 200 scores for use in the regression analysis. Full function computescore.m is shown in Listing B.11.

Before implementing the general linear model computations (the regression analysis), the scores struct is reorganized for convience by the code line shown in Listing 3.4. The reorganization stacks the 10×20 sized score struct such that the 20 columns are stacked below each other a new 200×1 sized struct called `collect`.

```
collect = score(:);
```

Listing 3.4: Code for reorganizing the scores struct.

For computing the p-values and corresponding β coeffiecients described in Section 2.4, the call shown in Listing 3.5 is done. The full function `newglm.m` with header is shown in Listing B.12.

```
for i = 1:size(responses.Δ,2)
    for j = 1:size(data,1)
        stats{i,j} = regstats(responses.Δ(:,i),data{j},'linear', ...
            {'fstat','rsquare','beta'});
        pvals(i,j) = stats{i,j}.fstat.pval;
        betas(i,j) = stats{i,j}.beta(2);
        rsquares(i,j) = stats{i,j}.rsquare;
    end
end
```

Listing 3.5: Main code for calling regstats for use when carrying out the regression analysis. Full function code is shown in Listing B.12.

A simple method of thresholding is used to collect the p-values that are significant at $10\%, 5\%, 1\%$ and 0.1% levels. The code for doing this is shown in Listing 3.6.

```
significancelevel = [.1 .05 .01 .001];
for i = 1:length(significancelevel)
    signif{i} = find(pvals < significancelevel(i));
end
```

Listing 3.6: Code for identifying significant p values.

3.3.4 Visualization and plotting of deformation modes

For computing and visualizing the deformation modes as described in Section 2.3.1, the MATLAB script `defmodes.m` is used. A small part of the code is shown in Listing 3.7 and signifies the computation done in Equation 2.8. The index i from the equation ranges from 1 to $200 = c \times r$ in which c is the number of sparse principal components and r is the number of stop values. The indices i and j in the code part are different and signify the c and r indices and therefore, have the ranges $i = 1, \cdots 20$ and $j = 1, \cdots 10$. The full script `defmodes.m` for computing and visualizing deformation modes for all 200 principal components is shown in Listing B.13.

```
LMp.bl_p(:,i) = normed + std*sqrt(abs(spca.sv(j).k10(i)) ...
    *abs(spca.sl(j).k10(:,i)));
LMp.bl_m(:,i) = normed - std*sqrt(abs(spca.sv(j).k10(i)) ...
    *abs(spca.sl(j).k10(:,i)));
```

Listing 3.7: Part of code for computing the deformation modes as described in Equation 2.8.

Chapter 4

Results and Evaluation

In this chapter the results from the analysis are presented and evaluated.

4.1 Results

The following description concerns four deformation mode figures in the present Section. All four mentioned figures share these properties: blue shape represents mean corpus callosum shape. Green and red shapes represent deformation modes for $s = \pm 1$ as described in Equation 2.8. They also contain data for principal components 1 to 10, but with number of non-zero variables ranging from 2 (extremely sparse) to 156 (regular non-sparse principal component analysis):

- Figure 4.1. Baseline mean shape. Number of non-zero variables ranges from 2 to 75.
- Figure 4.2. Baseline mean shape. Number of non-zero variables ranges from 83 to 156.
- Figure 4.3. Follow-up mean shape. Number of non-zero variables ranges from 2 to 75.

- Figure 4.4. Follow-up mean shape. Number of non-zero variables ranges from 83 to 156.

Table 4.1 shows an overview of the significant score and for which clinical variables they are significant to and also, at what significance level. Refer to tables in Appendix C for exact, corresponding β and p-values.

Tables C.1, C.2, C.3, C.4 and C.5 contain the full list of all β coefficients computed in the regression analysis described in Section 2.4. Tables C.6, C.7, C.8, C.9 and C.10 contain the corresponding p-values.

4.2 Evaluation

The following is an interpretation of the physical manifestations, or deformation modes of mean shapes, depicted in Figures 4.1, 4.2, 4.3 and 4.4, based on the outcome of the regression analysis results in Table 4.1:

MEMORY The fact that the analysis shows no significance between the principal scores and this clinical variable, is a bit discouraging. However, according to [2], their analysis of the mental processing assessments indicate that the associated corpus callosum changes might be diffuse and therefore, hard to detect by use of SPCA.

SPEED The analysis shows the highest significance of all the clinical variables. Ryberg et al. [3] found that the gait speed was associated with overall corpus callosum atrophy as well as reductions in CC1, CC2 and CC5 (see Figure 1.1 for subdivisions). The most significant score, PC6 for 148 non-zero variables actually seem to encompass the same mentioned areas. The same can be said for the scores PC4 (n equal to 140, 148 and 156), and in particular, they seem to point towards changes in the subregion CC2 (rostral body). PC5 (n equal to 132) and PC6 (n equal to 156) seem similar to the changes of PC4. The last significant deformation mentioned here is the most sparse, PC6 (n equal to 10), also seems to explain part of the deformation in CC2 (rostral body).

EXECUTIVE Jokinen et al. [2] found significant correlation between atrophy of CC1 region and scores for executive motor assessments. The only significant score for this clinical variable is that of PC10 (n equal to 156, which is full PCA), shows deformation modes that cover CC1, but also a

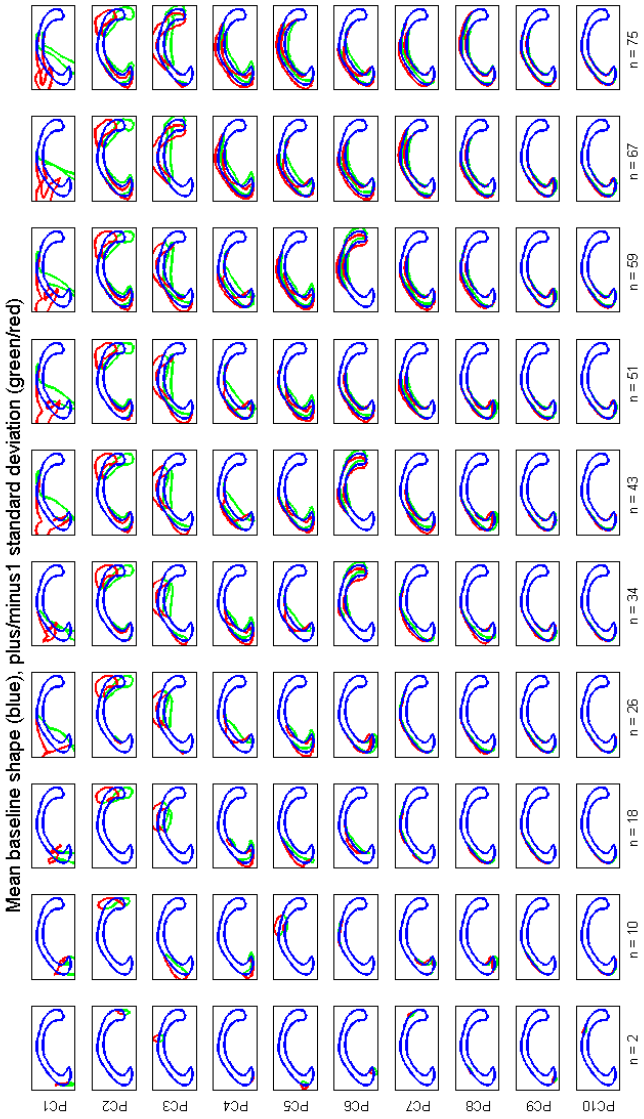

Figure 4.1: Deformation modes for PC1 to PC10, number of non-zero variables from 2 to 75. Blue represents baseline mean shape. Green and red represent plus 1 and minus 1 standard perturbed deformation mode as of Equation 2.8.

Figure 4.2: Deformation modes for PC1 to PC10, number of non-zero variables from 83 to 156 (regular PCA). Blue represents baseline mean shape. Green and red represent plus 1 and minus 1 standard perturbed deformation mode as of Equation 2.8.

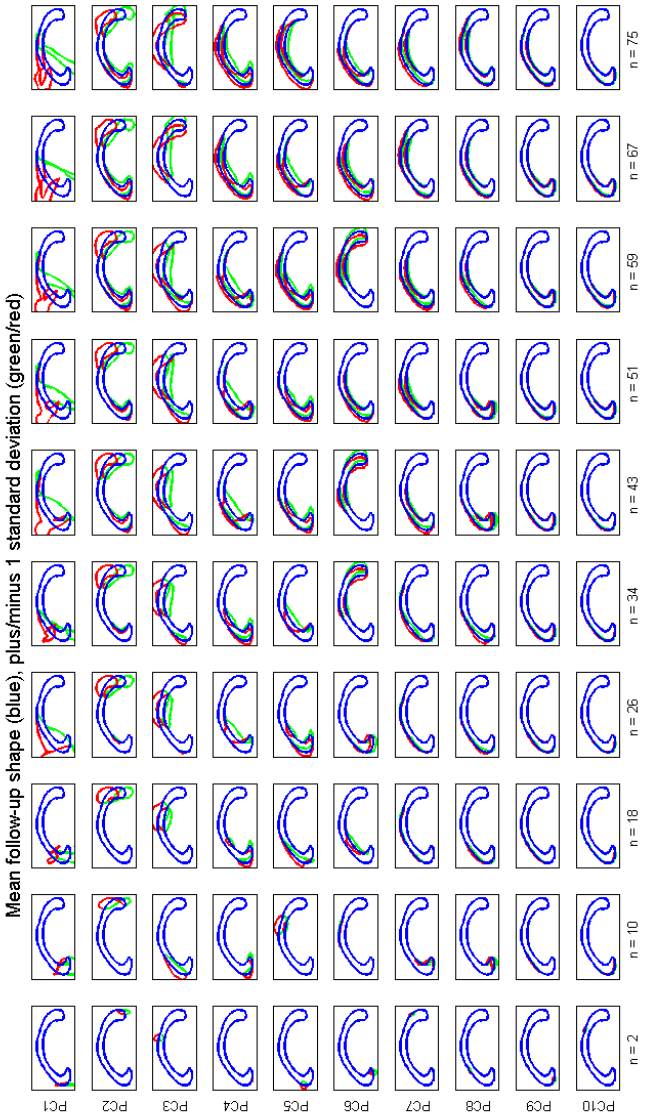

Figure 4.3: Deformation modes for PC1 to PC10, number of non-zero variables from 2 to 75. Blue represents follow-up mean shape. Green and red represent plus 1 and minus 1 standard perturbed deformation mode as of Equation 2.8.

Figure 4.4: Deformation modes for PC1 to PC10, number of non-zero variables from 83 to 156 (regular PCA). Blue represents follow-up mean shape. Green and red represent plus 1 and minus 1 standard perturbed deformation mode as of Equation 2.8.

small part of CC2 (rostral body) and CC5 (splenium). The present analysis thus points towards executive motor performance may manifest itself in all these three CC subdivisional regions.

verbal Jokinen et al. [2] found significance between this clinical parameter and CC atrophy in the overall CC as well as CC4 (isthmus) subregion. The present analysis clearly show an overal change in the CC shape, but also large changes in CC1, CC3 and CC5. Jokinen et al. expected to see a change in the anterior part which is actually evident in the present analysis. However, at a 10 percent significance level and with relative low sparsity (n equal to 99, 107 and 156 for PC3), the results seem to point more towards an overall CC shape change explanation than that of specific, local changes.

gdstotal Ryberg et al. [3] found no significance between the geriatric depression scale assessments and local corpus callosum area changes. The present analysis has found a 10 percent significance of an overall corpus atrophy may be explained by the changes in this clinical variable.

clinical variable	MEMORY	SPEED	EXECUTIVE	verbal	GDSTOTAL
significance level					
$p < 10$ %	none	PC1, n=2		PC3, n=99	PC3, n=99
		PC1, n=43		PC3, n=107	PC3, n=107
		PC5, n=2		PC3, full PCA	PC3, n=115
		PC5, n=91			
		PC7, n=18			
		PC7, n=26			
		PC7, n=34			
		PC7, n=99			
		PC7, n=107			
		PC8, n=59			
		PC9, n=107			
		PC9, n=132			
		PC9, n=140			
$p < 5$ %	none	PC4, n=140		none	none
		PC4, n=148			
		PC4, full PCA			
		PC5, n=132			
		PC6, n=10			
		PC6, full PCA			
$p < 1$ %	none	PC6 (n =148)	PC10, full PCA	none	none
$p < 0.1$ %	none	none	none	none	none

Table 4.1: Overview of which scores are significant to which clinical parameters and at what significance level.

Chapter 5

Discussion

The three main foci of the analytical work contained within the thesis work were to:

- Perform a sparse principal component analysis on treated landmarks to make a sparse representation of local corpora callosa contour changes in the mid-sagittal perspective of the human brain.
- Perform a regression analysis between the derived variables and changes in clinical performance assessments prepared from the LADIS study.
- Visualize the local corpus callosum shape changes deemed significant by the spca and regression analysis and evaluate on their interpretability with respect to previous analytical work presented in research articles by LADIS associated crew.

When comparing the results with those of Jokinen et al. [2] and Ryberg et al. [3], some of the significant local corpus shape changes that were found seemed to correspond with the results of these groups. Sjöstrand et al. [5], Sjöstrand et al. [6] and Sjöstrand [4] already performed analyses with the same foci and concluded that a sparse representation of the variables worked well.

A further analysis along a similar path could include area computations of the difference between deformation modes and mean shapes to accomodate with research that specifically focuses on local corpus callosum area changes.

Chapter 6

Conclusion

A extensive amount of available data from the LADIS Leukoaraiosis And DISability) Study was provided for this thesis.

After performing three steps of narrowing down the number of test persons, a full baseline and follow-up data set consisting of bitmap images, associated expert reviewed landmarks of the corpus callosum contour outline and 5 clinical parameters were ready for analysis.

A sparse principal component analysis was performed on the landmark data with the aim of detecting local corpus callosum shape changes signifying local atrophy. A subsequent regression analysis performed between the outcome of this analysis and the clinical parameters showed results that to some extent correspond acceptably with the results of the found literature on conductions of similar studies.

Appendix A

Additional Figures

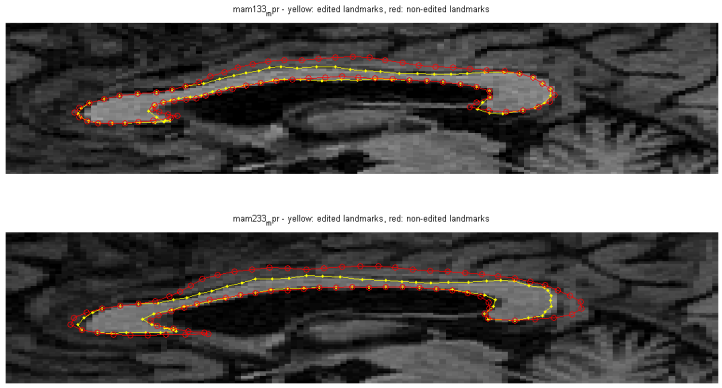

Figure A.1: Top: close-up of CC baseline MR scan of test person AM33. Red shape represents landmarks computed by the learning-based active appearance model and yellow shape represents the expert corrected landmarks. Bottom: same as top for the follow-up scan. This test person was not sorted away in reduction step 3 described in Chapter 3.3.1.

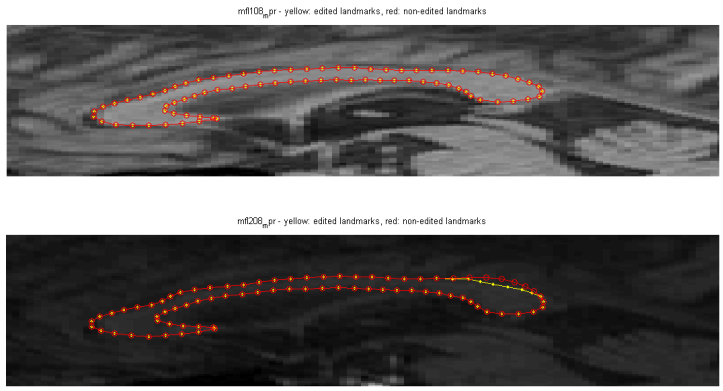

Figure A.2: Top: close-up of CC baseline MR scan of test person FL08. Red shape represents landmarks computed by the learning-based active appearance model and yellow shape represents the expert corrected landmarks. Bottom: same as top for the follow-up scan. This test person was not sorted away in reduction step 3 described in Chapter 3.3.1.

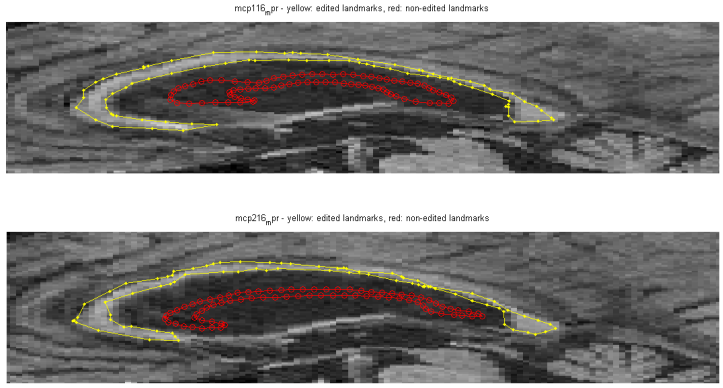

Figure A.3: Top: close-up of CC baseline MR scan of test person CP16. Red shape represents landmarks computed by the learning-based active appearance model and yellow shape represents the expert corrected landmarks. Bottom: same as top for the follow-up scan. This test person was sorted away in reduction step 3 described in Chapter 3.3.1.

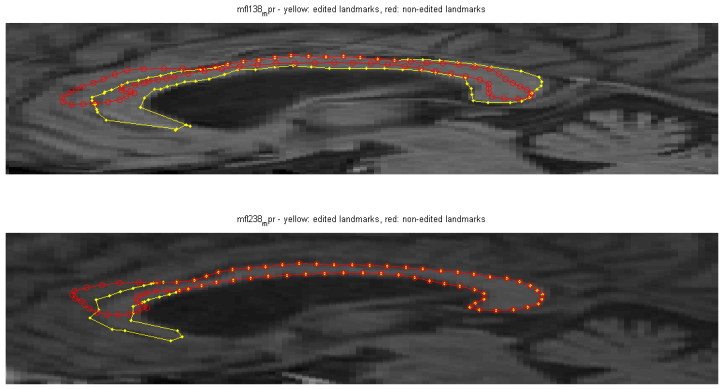

Figure A.4: Top: close-up of CC baseline MR scan of test person FL38. Red shape represents landmarks computed by the learning-based active appearance model and yellow shape represents the expert corrected landmarks. Bottom: same as top for the follow-up scan. This test person was sorted away in reduction step 3 described in Chapter 3.3.1.

Appendix B

Matlab Code Listings

```
%% Main script for data extraction

% Import clinical variables by running script clin.m:
load clinimp

% Identify test persons from clin.name list who have both ...
    baseline and follow-up scans and return lists referring to ...
    nomenclature of both the .bmp and landmark folder as well as ...
    of the LADIS Access database
[imgname clin LMi data Data x] = convertname(clin);

% Run script inspect.m to sort away erronous landmarks and ...
    reduce LMi. This has already been done and new LMi and ...
    sortindex is saved under reduction_03
load reduction_03;

% Run function clinred_02.m with new LMi to get updated clin
clin = clinred_02(clin,sortindex);

% Run function getcoords.m with data and new LMi to get ...
    non-erroneous landmark coordinates
x = getcoords(data,LMi);

% Compute SPCA on the normalized, edited landmark coordinates
maxiter = 150;
trace   = 1;
lambda  = 1;
```

```
stop    = -round(linspace(2,156,20));
K       = 10;
% for i = 1:length(stop)
%     [a b c d ¬] = spca(x.ed.Δ_norm', [], K, ...
%         lambda, stop(i), maxiter, trace);
%     SPCA.sl.K10(i).norm = a;
%     SPCA.sv.K10(i).norm = b;
%     SPCA.pcal.K10.norm  = c;
%     SPCA.pcav.K10.norm  = d;
% end

% The SPCA computations takes approx. 1,5 hours on a regular ...
    laptop and resulting SPCA struct has been saved under ...
    spca_final_K10_ed (wrong toc value)
load spca_final_K10_ed;

% Reorganize structure of SPCA and put in spca struct
for i = 1:length(stop)
    spca.sl(i).k10 = SPCA.sl.K10(i).norm;
    spca.sv(i).k10 = SPCA.sv.K10(i).norm;
end
spca.pcal = SPCA.pcal.K10.norm;
spca.pcav = SPCA.pcav.K10.norm;

% Compute scores
[score s] = computescore(spca,clin,x);

% Reorganize scores (stack columns under each other)
collect = score(:);

% Compute p-values and betas (regression analysis)
[pvals betas] = newglm(clin,collect);

% Find significant p-values
significancelevel = [.1 .05 .01 .001];
for i = 1:length(significancelevel)
    signif{i} = find(pvals < significancelevel(i));
end

% Load spca and x into workspace before plotting deformation ...
    modes using the script defmodes.m.
```

Listing B.1: Main Matlab script for implementation of the analysis.

```
% dirs.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Description:  Scripts sets the base directories for use when ...
    extracting clinical variables from Excel datasheets, bitmap ...
    images and landmark coordinates.
%
% Author:       Nicolas Tiaki Otsu (s072254@student.dtu.dk)
% Last edited:  June 13, 2011
%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Set base directories
base.am='E:\LADIS\ccam\';
dir_bmp.am=dir([base.am '*.bmp']);
dir_mat.am=dir([base.am '*.mat']);
base.cp='E:\LADIS\cccp\';
dir_bmp.cp=dir([base.cp '*.bmp']);
dir_mat.cp=dir([base.cp '*.mat']);
base.ladis='E:\LADIS\ccladis\';
dir_bmp.ladis=dir([base.ladis '*.bmp']);
dir_mat.ladis=dir([base.ladis '*.mat']);

base.excel='E:\LADIS\Excel\';

save('dirs','dir_bmp','dir_mat','base')
```

Listing B.2: Matlab code for setting file directories.

```
% clinimp.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Description:  Script extracts clinical variables from LADIS
%               database Excel sheets. Detects test persons with
%               all associated clinical variables and saves data in
%               clinimp.mat.
%
% Author:       Nicolas Tiaki Otsu (s072254@student.dtu.dk)
% Last edited:  June 4, 2011
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Load directories for data extraction (dirs.m):
load dirs

% Import 'MEMORY', 'MEM3y', 'SPEED', 'SPEED3y', 'EXECUTIVE', ...
    'EXEC3y':
[full.num.compound_measures_wp4,full.txt.compound_measures_wp4,¬] ...
    = ...
    xlsread([base.excel 'compound_measures_wp4']);
order     = [1 12 2 11 3 10];
clin.vars = full.txt.compound_measures_wp4(1,order+1);
clin.num  = full.num.compound_measures_wp4(:,order);

% Import 'verbal', 'verbal3y', 'gdstotal', 'gdstotal3y':
[full.num.table2_baseline,full.txt.table2_baseline,¬] = ...
    xlsread([base.excel 'table2_baseline.xls']);
[full.num.table2_3y,full.txt.table2_3y,¬] = ...
    xlsread([base.excel 'table2_3y.xls']);

% Correct full.num.table2_3y due to xlsread not properly ...
    inserting AM2, AM3 and AM4:
full.num.table2_3y = [repmat(str2num('NaN'),3,207); ...
    full.num.table2_3y];
```

```
order_bl = ...
    [find(ismember(full.txt.table2_baseline(1,:),'verbal')) ...
    find(ismember(full.txt.table2_baseline(1,:),'gdstotal'))];
order_fu = [find(ismember(full.txt.table2_3y(1,:),'verbal3y')) ...
    find(ismember(full.txt.table2_3y(1,:),'gdstotal3y'))];

% Update clin.vars and clin.num:
for i = 1:2
    clin.vars = [clin.vars ...
        full.txt.table2_baseline(1,order_bl(i)) ...
        full.txt.table2_3y(1,order_fu(i))];
    clin.num = [clin.num ...
        full.num.table2_baseline(:,order_bl(i)-1) ...
        full.num.table2_3y(:,order_fu(i)-1)];
end

% Import 'sex', 'birthday', 'daterif', 'datefu3':
[full.num.table1_baseline,full.txt.table1_baseline,~] = ...
    xlsread([base.excel 'table1_baseline.xls']);
[full.num.table1_3y,full.txt.table1_3y,raw] = ...
    xlsread([base.excel 'table1_3y.xls']);

sexcol     = find(ismember(full.txt.table1_baseline(1,:),'sex'));
male       = ismember(full.txt.table1_baseline(2:end,sexcol),'M');
fema       = ~male;
bdaycol    = ...
    find(ismember(full.txt.table1_baseline(1,:),'birthday'));
daterifcol = ...
    find(ismember(full.txt.table1_baseline(1,:),'daterif'));
datefu3col = find(ismember(full.txt.table1_3y(1,:),'datefu3'));
agebl      = computeage(full.txt.table1_baseline(:,bdaycol), ...
    full.txt.table1_baseline(:,daterifcol));
agefu      = computeage(full.txt.table1_baseline(:,bdaycol), ...
    full.txt.table1_3y(:,datefu3col));

% Update clin.vars and clin.num
clin.vars  = [clin.vars 'age' 'age3y' 'male' 'female'];
clin.num   = [clin.num agebl agefu male fema];

% Find persons for whom all vars exists for both bl and fu:
index     = find(sum(~isnan(clin.num)')==length(clin.vars));
clin.num  = clin.num(index,:);
clin.name = full.txt.compound_measures_wp4(index,1);

save('clinimp','clin','full')
```

Listing B.3: Matlab code for extracting clinical variables from the LADIS Excel datasheets and collect test persons who have all selected clinical variable data.

```
function age = computeage(bday, cday)

% computeage.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
```

```
% Description:  Function for computing the age as difference ...
    between two given dates.
%
% Call:         AGE = COMPUTEAGE(BDAY, CDAY)
%
% Input:        BDAY, birthday cell in the form (n,dd-mm-yyyy) ...
    with n
%               being the number of observations.
%               CDAY, checkdate cell in the form (n,dd-mm-yyyy).
%
% Output:       AGE, double (n,1) vector with n computed ages.
%
% Author:       Nicolas Tiaki Otsu (s072254@student.dtu.dk)
% Last edited:  June 4, 2011
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

age = zeros(size(bday));
for i = 2:length(age)
    if sum(cday{i}) ~= 0
        if str2double(cday{i}(4:5)) > str2double(bday{i}(4:5))
            age(i) = str2double(cday{i}(7:10)) - ...
                str2double(bday{i}(7:10));
        end
        if str2double(cday{i}(4:5)) < str2double(bday{i}(4:5))
            age(i) = str2double(cday{i}(7:10)) - ...
                str2double(bday{i}(7:10)) - 1;
        end
        if str2double(cday{i}(4:5)) == str2double(bday{i}(4:5))
            if str2double(cday{i}(1:2)) < str2double(bday{i}(1:2))
                age(i) = str2double(cday{i}(7:10)) - ...
                    str2double(bday{i}(7:10)) - 1;
            else
                age(i) = str2double(cday{i}(7:10)) - ...
                    str2double(bday{i}(7:10));
            end
        end
    else
        age(i) = str2double(cday{i});
    end
end
age = age(2:end);
```

Listing B.4: Matlab function for computing the test person ages.

```
function [imgname clin LMi data Data x] = convertname(clin)

% convertname.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Description:  Function converts test person names from the ...
    LADIS Access database nomenclature to the nomenclature used ...
    for the .bmp and edited corpus callosum contour landmark ...
    coordinates. Function also corrects for minor ...
```

```
      inconsistencies in the nomenclature. Finally, function sorts ...
      away those test person names that do not have both baseline ...
      and follow-up .bmp images and coordinates associated with them.
6  %
7  % Call:         [IMGNAME CLIN LMI DATA1 DATA2 X] = CONVERTNAME(CLIN)
8  %
9  % Input:        CLIN, (n,1) struct with field 'NAME' consisting ...
      of strings with test person codes of the form 'XXnn', in ...
      which XX signifies the hospital code and nn, the person ...
      number. n signifies the number of observations. The form ...
      'XXnn' obeys the nomenclature for the LADIS Access database.
10 %
11 % Output:       IMGNAME, (n,2) struct with person number names ...
      obeying the nomenclature for the .bmp images and edited ...
      corpus callosum contour landmark coordinates. First column ...
      is baseline name, second column is follow-up name.
12 %               CLIN, struct with the 'NAME' field adjusted to ...
      correspond with the test person names in imgname. The ...
      remained outputs are passed on via the functions that are ...
      called by convertname.m
13 %
14 % Author:       Nicolas Tiaki Otsu (s072254@student.dtu.dk)
15 % Last edited:  June 4, 2011
16 %
17 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
18
19 load dirs
20
21 matches = lower(clin.name);
22
23 centers = {'am';'cp';'fl';'gr';'gt';'he';'hu';'ls';'ma';'nc';'pa'};
24 % Important note: There does not exist fu scans for nc ...
      (centers(10))!
25
26 % Load data from all 11 centers into data
27 for i = 1:86
28     data(i)=load([base.am dir_mat.am(i).name]);
29         Data.am(i)=load([base.am dir_mat.am(i).name]);
30 end
31 for i = 1:107
32     data(end+1)=load([base.cp dir_mat.cp(i).name]);
33         Data.cp(i)=load([base.cp dir_mat.cp(i).name]);
34 end
35 cent.bl = zeros(785+length(data),length(matches));
36 cent.fu= zeros(785+length(data),length(matches));
37 for i = 1:785
38     data(end+1)=load([base.ladis dir_mat.ladis(i).name]);
39     Data.ladis(i)=load([base.ladis dir_mat.ladis(i).name]);
40 end
41
42 % Add second column to matches
43 for k = 1:11
44     for i = 1:length(matches)
45         if findstr(char(matches(i)),char(centers(k)))
46             matches{i,2} = k;
```

```
47          elseif matches{i,2} < 1
48              matches{i,2} = 0;
49          end
50      end
51  end
52
53  for i = 1:length(data)
54      for j = 1:size(matches,1)
55          if length(matches{j,1}) == 3
56              tmp = [matches{j,1}(1:2) '0' matches{j,1}(3)];
57          else if length(matches{j,1}) == 5
58                  tmp = [matches{j,1}(1:2) matches{j,1}(4:5)];
59              else if length(matches{j,1}) == 4
60                      tmp = matches{j,1};
61                  end
62              end
63          end
64          if findstr(data(i).basename(2:6),[tmp(1:2) '1' tmp(3:4)])
65              cent.bl(i,j) = matches{j,2};
66          end
67          if findstr(data(i).basename(2:6),[tmp(1:2) '2' tmp(3:4)])
68              cent.fu(i,j) = matches{j,2};
69          end
70      end
71  end
72
73  % Locate rows and cols for persons with matches for all vars ...
        (.full) and for individual hospitals (.part)
74  [row.bl.full col.bl.full] = find(cent.bl);
75  for i = 1:length(centers)
76      [row.bl.part{i} col.bl.part{i}] = find(cent.bl == i);
77  end
78
79  [row.fu.full col.fu.full] = find(cent.fu);
80  for i = 1:length(centers)
81      [row.fu.part{i} col.fu.part{i}] = find(cent.fu == i);
82  end
83
84  % Find matching stringnames for baseline and follow-up and ...
        update imgname
85  k = 0;
86  newclin.name = [];
87  for i = 1:length(row.bl.full)
88      for j = 1:length(row.fu.full)
89          if findstr(data(row.bl.full(i)).basename([2:3 5:6]), ...
90                  data(row.fu.full(j)).basename([2:3 5:6]))
91              newclin.name{end+1} = ...
                    data(row.bl.full(i)).basename([2:3 5:6]);
92              k = k + 1;
93              imgname{k,1} = data(row.bl.full(i)).basename;
94              imgname{k,2} = data(row.fu.full(j)).basename;
95          end
96      end
97  end
98
```

```
% Update clin to contain only test persons who correspond to ...
    those in imgname and collect .bmp and landmark indices to ...
    use for SPCA
[LMi clin x] = clinred(data,imgname,clin);
```

Listing B.5: Matlab function for converting test person names from LADIS Excel datasheet nomenclature to that of the bitmap and mat file names. Function corrects for minor inconsistencies in the naming pattern of the Excel datasheets and collects data for those test person names who have both baseline and follow-up scans made.

```
function [LMi clin x] = clinred(data,imgname,clin)

% clinred.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Description:  Function for reducing CLIN so that the test ...
    person names correspond to those of imgname. Function also ...
    returns LMI, a struct holding the indexes to be used for ...
    collecting matching .bmp and edited corpus callosum contour ...
    landmark coordinates.
%
% Call:         [LMI CLIN X] = CLINRED(DATA,IMGNAME,CLIN)
%
% Input:        DATA, struct containing field 'basename' with ...
    string of test person name.
%               IMGNAME, (n,2) cell containing strings of test ...
    person names. First column is baseline, second, follow-up.
%               CLIN, struct with double type field 'NUM' of ...
    size (m,p) with clinical observation data and with m>n being ...
    the unreduced number of test persons and p being the number ...
    of clinical variables. CLIN also holds cell type field ...
    'NAME' of size (m,1) with m unreduced test person names.
%
% Output:       LMI, struct with fields 'AM', 'CP', 'LADIS' and ...
    'FULL' containing indexes to be used to identify in which ...
    folders which .bmp and edited corpus callosum contour ...
    landmark coordinates are found. Remaining outputs are passed ...
    on by the functions that are called by clinred.m
%
% Author:       Nicolas Tiaki Otsu (s072254@student.dtu.dk)
% Last edited:  June 6, 2011
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

load dirs
matches = lower(clin.name);
centers = {'am';'cp';'fl';'gr';'gt';'he';'hu';'ls';'ma';'nc';'pa'};

% Create LMi index struct
LMi.am = 0; LMi.cp = 0; LMi.ladis = 0;
for i = 1:length(imgname)
    for j = 1:length(data)
```

```
28          for k = 1:2
29              if sum(findstr(imgname{i,k}, data(j).basename))
30                  if findstr(imgname{i,k}(2:3), centers{1})
31                      LMi.am(end+1,k) = j;
32                  end
33                  if findstr(imgname{i,k}(2:3), centers{2})
34                      LMi.cp(end+1,k) = j;
35                  end
36                  if ¬sum(findstr(imgname{i,k}(2:3), centers{1})) ...
                        && ¬sum(findstr(imgname{i,k}(2:3), centers{2}))
37                      LMi.ladis(end+1,k) = j;
38                  end
39              end
40          end
41      end
42  end
43
44  LMi.am = LMi.am(2:end,:); LMi.cp = LMi.cp(2:end,:); LMi.ladis = ...
        LMi.ladis(2:end,:);
45  LMi.full = [LMi.am; LMi.cp; LMi.ladis];
46
47  % Remove zero values in LMi struct fields
48  LMitemp = LMi;
49  LMitemp.am(LMitemp.am == 0)       = [];
50  LMitemp.cp(LMitemp.cp == 0)       = [];
51  LMitemp.ladis(LMitemp.ladis == 0) = [];
52  LMitemp.full(LMitemp.full == 0)   = [];
53
54  LMi.am    = reshape(LMitemp.am,length(LMi.am)/2,2);
55  LMi.cp    = reshape(LMitemp.cp,length(LMi.cp)/2,2);
56  LMi.ladis = reshape(LMitemp.ladis,length(LMi.ladis)/2,2);
57  LMi.full  = reshape(LMitemp.full,length(LMi.full)/2,2);
58
59  % Adjust indeces of LMi.cp and LMi.ladis to match with folder ...
        indeces
60  LMi.cp    = LMi.cp - length(dir_bmp.am);
61  LMi.ladis = LMi.ladis - length(dir_bmp.am) - length(dir_bmp.cp);
62
63  % Use the baseline-follow-up matches in imgname to reduce the ...
        test person names and data in clin.name and clin.num to ...
        appropriately match
64  comp = 0;
65  for i = 1:length(LMi.full)
66      for j = 1:length(matches)
67          if length(matches{j,1}) == 3
68              tmp = [matches{j,1}(1:2) '0' matches{j,1}(3)];
69          else if length(matches{j,1}) == 5
70                  tmp = [matches{j,1}(1:2) matches{j,1}(4:5)];
71              else if length(matches{j,1}) == 4
72                      tmp = matches{j,1};
73                  end
74              end
75          end
76          if findstr(data(LMi.full(i,1)).basename([2:3 5:6]), tmp)
77              comp(end + 1) = j;
```

```
78          end
79      end
80  end
81
82  comp      = comp(2:end);
83  clin.name = clin.name(comp,:);
84  clin.num  = clin.num(comp,:);
85
86  % Create struct x holding both edited and non-edited landmark ...
        coordinates for the test person indeces in LMi.full
87  x = getcoords(data,LMi);
88
89  % Assuming that last four columns of clin.vars are 'age', ...
        'age3y', 'male', 'female', subtract follow-up data from ...
        baseline data for all other columns and store in clin.Δ. ...
        Also create '(diff)' variable names in clin.dvars
90  clin = Δclin(clin);
```

Listing B.6: Matlab function responsible for collecting those test persons from the Excel datasheets that match with those of the bitmap and mat files. Function also returns LMi which holds indeces to be used for collecting matching bitmap images and coordinates.

```
1   function x = getcoords(data,LMi)
2
3   % getcoords.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4   %
5   % Description:  Function for extracting both edited and ...
        non-edited corpus callosum contour landmark coordinates.
6   %
7   % Call:         X = GETCOORDS(DATA,LMI)
8   %
9   % Input:        DATA, struct containing field 'basename' with ...
        string of test person name.
10  %               LMI, struct with fields 'AM', 'CP', 'LADIS' and ...
        'FULL' containing indexes to be used to identify in which ...
        folders which .bmp and edited corpus callosum contour ...
        landmark coordinates are found.
11  %
12  % Output:       X, struct holding the landmark coordinates ...
        retrieved from DATA and chosen by the test person indeces in ...
        LMI.FULL.
13  %
14  % Author:       Nicolas Tiaki Otsu (s072254@student.dtu.dk)
15  % Last edited:  June 5, 2011
16  %
17  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
18
19  % Create struct x holding both edited and non-edited landmark ...
        coordinates for the test person indeces in LMi.full
20  for i = 1:length(LMi.full)
21      x.ed.bl(:,i) = [data(LMi.full(i,1)).landmarks_edited(:,1); ...
22          data(LMi.full(i,1)).landmarks_edited(:,2)];
```

```
23      x.ed.fu(:,i) = [data(LMi.full(i,2)).landmarks_edited(:,1); ...
24          data(LMi.full(i,2)).landmarks_edited(:,2)];
25      x.ed.Δ_norm(:,i) = normalize(x.ed.bl(:,i)) - ...
26          normalize(x.ed.fu(:,i));
27
28      x.ned.bl(:,i) = [data(LMi.full(i,1)).landmarks(:,1); ...
29          data(LMi.full(i,1)).landmarks(:,2)];
30      x.ned.fu(:,i) = [data(LMi.full(i,2)).landmarks(:,1); ...
31          data(LMi.full(i,2)).landmarks(:,2)];
32      x.ned.Δ_norm(:,i) = normalize(x.ned.bl(:,i)) - ...
33          normalize(x.ned.fu(:,i));
34  end
```

Listing B.7: Matlab function for collecting landmark coordinates contained in the data struct via the indeces stored in LMi.

```
1   function clin = Δclin(clin)
2
3   % Δclin.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4   %
5   % Description:  Function for computing the change in clinical ...
        performance assessments from baseline to follow-up.
6   %
7   % Call:         CLIN = DELTACLIN(CLIN)
8   %
9   % Input:        CLIN, struct with double type field 'NUM' of ...
        size (n,p) with clinical performance data of n test persons ...
        and p  clinical variables. CLIN also holds cell type field ...
        'VARS' of size (1,p) with clinical variable names.
10  %
11  % Output:       CLIN, same as input, but with two extra fields ...
        added. 'DELTA', double type of size (n,k), holds the ...
        computed changes in the performance data after centering. ...
        'DVARS', double type of size (1,k), holds the names of the k ...
        variable names. k has size k = p/2-4 due to the assumption ...
        that the last four columns of CLIN.VARS are 'AGE', 'AGE3y', ...
        'MALE', 'FEMALE', which are variables that do not need their ...
        baseline-follow-up differences computed.
12  %
13  % Author:       Nicolas Tiaki Otsu (s072254@student.dtu.dk)
14  % Last edited:  June 14, 2011
15  %
16  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
17
18  tmp = clin.num(:,1:(length(clin.vars)-4));
19  for i = 1:(size(tmp,2))/2
20      clin.Δ(:,i) = center(tmp(:,2*i)) - center(tmp(:,2*i-1));
21      clin.dvars(i)= {[clin.vars{2*i-1} '(diff)']};
22  end
```

Listing B.8: Matlab function for computing the difference in the clinically assessed performance parameters from baseline to follow-up.

```
% inspect.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Description:  Script sort away test persons with erroneous ...
    edited landmarks associated with their MR scans.
%
% Author:       Nicolas Tiaki Otsu (s072254@student.dtu.dk)
% Last edited:  June 6, 2011
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

load dirs

folders = {'am','cp','ladis'};
axisstep = .2;
for i = 1:length(folders);
    for j = 1:length(LMi.(char(folders(i))))
        %         if ¬imgsort_01_cell{i,j}

        % Load corresponding baseline and follow-up images for ...
            comparance
        Ibl = imread([base.(char(folders(i))) ...
            dir_bmp.(char(folders(i))) ...
    (LMi.(char(folders(i)))(j,1)).name]);
        Ifu = imread([base.(char(folders(i))) ...
            dir_bmp.(char(folders(i))) ...
        (LMi.(char(folders(i)))(j,2)).name]);

        LM.bl = ...
            Data.(char(folders(i)))(LMi.(char(folders(i)))(j,1));
        LM.fu = ...
            Data.(char(folders(i)))(LMi.(char(folders(i)))(j,2));

        % Plot baseline image with CC landmarks (both edited and ...
            non)
        subplot(2,1,1)
        imagesc(Ibl), colormap gray, hold on
        plot([LM.bl.landmarks_edited(:,1); ...
            LM.bl.landmarks_edited(1,1)], ...
            [(LM.bl.landmarks_edited(:,2)+1); ...
                (LM.bl.landmarks_edited(1,2)+1)],'.-y')
         plot([LM.bl.landmarks(:,1); LM.bl.landmarks(1,1)], ...
            [(LM.bl.landmarks(:,2)+1); ...
                (LM.bl.landmarks(1,2)+1)],'o-r')
        title([LM.bl.basename ...
            ' - yellow: edited landmarks, red: non-edited ...
                landmarks'])

        axis([(1-axisstep)*min(LM.bl.landmarks_edited(:,1)) ...
            (1+axisstep)*max(LM.bl.landmarks_edited(:,1)) ...
            (1-axisstep)*min(LM.bl.landmarks_edited(:,2)) ...
            (1+axisstep)*max(LM.bl.landmarks_edited(:,2))])
        axis off

        % Plot follow-up image with CC landmarks (both edited ...
```

```
            and non)
46          subplot(2,1,2)
47          imagesc(Ifu), colormap gray, hold on
48            plot([LM.fu.landmarks_edited(:,1); ...
                  LM.fu.landmarks_edited(1,1)], ...
49              [(LM.fu.landmarks_edited(:,2)+1); ...
                  (LM.fu.landmarks_edited(1,2)+1)],'.-y')
50           plot([LM.fu.landmarks(:,1); LM.fu.landmarks(1,1)], ...
51              [(LM.fu.landmarks(:,2)+1); ...
                  (LM.fu.landmarks(1,2)+1)],'o-r')
52  %         plot(LM.fu.landmarks_edited(:,1), ...
53  %    LM.fu.landmarks_edited(:,2)+1,'.-y')
54  %         plot(LM.fu.landmarks(:,1),LM.fu.landmarks(:,2)+1,'o-r')
55          title([LM.fu.basename ...
56              ' - yellow: edited landmarks, red: non-edited ...
                  landmarks'])
57
58          axis([(1-axisstep)*min(LM.fu.landmarks_edited(:,1)) ...
59              (1+axisstep)*max(LM.fu.landmarks_edited(:,1)) ...
60              (1-axisstep)*min(LM.fu.landmarks_edited(:,2)) ...
61              (1+axisstep)*max(LM.fu.landmarks_edited(:,2))])
62          axis off
63
64          %          imgsort_01_cell{i,j} = waitforbuttonpress;
65          pause
66          clf
67      end
68  end
69  % end
70
71  % Save first sorting (only obviously non-erroneous landmarks ...
        accepted)
72  % save imgsort_01_cell imgsort_01_cell
73
74  load imgsort_01_cell
75  % Convert imgsort_0x from cell to numerical matrix
76  for i = 1:size(imgsort_01_cell,1)
77      for j = 1:size(imgsort_01_cell,2)
78          if imgsort_01_cell{i,j}
79              imgsort_01(i,j) = imgsort_01_cell{i,j};
80          else
81              imgsort_01(i,j) = 0;
82          end
83      end
84  end
85
86  % Reduce LMi according to sorting and create sortindex for later ...
        use when reducing number of clinical observations in clin
87  sortindex = 0;
88  for i = 1:size(imgsort_01,1)
89      k = size(LMi.(char(folders(i))),1);
90      for j = 1:size(imgsort_01,2)
91          if j ≤ k
92              sortindex(end+1) = imgsort_01_cell{i,j};
93              if ¬imgsort_01(i,j)
```

```
                LMi.(char(folders(i)))(j,:) = 0;
            end
        end
    end
end
sortindex = sortindex(2:end);

% Remove zero indeces in LMi and collect in LMi.full
LMi.am = reshape(LMi.am(LMi.am ≠ 0),sum(find(LMi.am ≠ 0) ...
    ./find(LMi.am ≠ 0))/2,2);
LMi.cp = reshape(LMi.cp(LMi.cp ≠ 0),sum(find(LMi.cp ≠ 0) ...
    ./find(LMi.cp ≠ 0))/2,2);
LMi.ladis = reshape(LMi.ladis(LMi.ladis ≠ 0),sum(find(LMi.ladis ≠...
     0) ...
    ./find(LMi.ladis ≠ 0))/2,2);
LMi.full = [LMi.am; LMi.cp; LMi.ladis];

% new LMi and sortindex is saved under reduction_03
```

Listing B.9: Matlab script for preparing for last reduction step of test persons based on sorting away erroneous edited corpus callosum contour landmarks that do not match the bitmap images based on visual inspection.

```
function clin = clinred_02(clin,sortindex)

% clinred_02.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Description:  Third function for reducing number of clinical ...
    variables.
%
% Call:         CLIN = CLINRED_02(CLIN,SORTINDEX)
%
% Input:        CLIN, struct with double type field 'NUM' of ...
    size (m,p) with m unreduced observations and p clinical ...
    variables, cell type field 'NAME' of size (m,1) with test ...
    person number according to LADIS Excel database ...
    nomenclature, and double type field 'DELTA' of size (m,p) ...
    containing computed baseline–follow–up differences in ...
    clinical observations.
%               SORTINDEX, double type of size (1,n) with n ...
    being the number of desired, reduced observations.
%
% Output:       CLIN, with mentioned fields with number of ...
    observations reduced from m to n.
%
% Author:       Nicolas Tiaki Otsu (s072254@student.dtu.dk)
% Last edited:  June 6, 2011
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

load dirs

j = 0;
```

```
clear num, clear Δ, clear name
for i = 1:size(clin.num,1)
    if sortindex(i)
        j = j + 1;
        num(j,:) = clin.num(i,:);
        name{j,:} = clin.name{i};
        Δ(j,:) = clin.Δ(i,:);
    end
end
clin.num = num;
clin.name = name;
clin.Δ = Δ;
```

Listing B.10: Matlab function for performing the last reduction step based on the sortindex list created by using the script inspect.m in Listing B.9.

```
function [score s] = computescore(spca,clin,x)

% computescore.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Description:  Function for computing sparse scores.
%
% Call:         [SCORE S] = COMPUTESCORE(SPCA,CLIN,X)
%
% Input:        SPCA, struct with fields SL with sparse ...
    loadings, SV, sparse loading vectors, PCAL, regular ...
    loadings, PCAV, regular loading vectors.
%               CLIN, struct with clinical variables.
%               X, struct with edited landmark coordinates.
%
% Output:       SCORE, cell of dimension (K, j) with K being the ...
    number of sparse principal components and j being the number ...
    of stop numbers. Each field is a double of the dimension ...
    (1,n) with n being the number of observations.
%               S, cell of dimension (1,K) containing the same ...
    information as SCORE. Each field is of dimensions (n, K)
%
% Author:       Nicolas Tiaki Otsu (s072254@student.dtu.dk)
% Last edited:  June 15, 2011
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

m = size(x.ed.Δ_norm,2);
c = size(spca.sl,2);
r = size(spca.sl(1).k10,2);
l = length(clin.dvars);
for i = 1:r
    for j = 1:c
        score{i,j} = spca.sl(j).k10(:,i)'*x.ed.Δ_norm;
    end
end

for i = 1:r
```

```
32      s{i}=reshape([score{:,i}],m,r);
33  end
```

Listing B.11: Matlab function computing the principal scores for use in the regression analysis as described in Section 3.3.3.

```
1   function [pvals betas] = newglm(responses,data)
2
3   % newglm.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4   %
5   % Description:  Function for implementing the regression ...
        analysis between the responses and data.
6   %
7   % Call:         [PVALS BETAS] = NEWGLM(RESPONSES,DATA)
8   %
9   % Input:        RESPONSES, struct with field 'DELTA' of type ...
        double and size (n,j) with n being the number of ...
        observations and j being the number of clinical variables.
10  %
11  %               DATA, cell of size (m, 1) with m being the ...
        number of principal components used for the analysis. Each ...
        cell entry is a double of size (1,n).
12  %
13  % Output:       PVALS, double of size (n,j) showing ...
        corresponding p-values for each entry in the RESPONSES.DELTA ...
        field.
14  %               BETAS, double of size (j,n) showing ...
        corresponding beta coefficients for each entry in the ...
        transposed RESPONSES.DELTA field.
15  %
16  % Author:       Nicolas Tiaki Otsu (s072254@student.dtu.dk)
17  % Last edited:  June 14, 2011
18  %
19  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
20
21  for i = 1:size(responses.∆,2)
22      for j = 1:size(data,1)
23          stats{i,j} = regstats(responses.∆(:,i),data{j},'linear', ...
24              {'fstat','rsquare','beta'});
25          pvals(i,j) = stats{i,j}.fstat.pval;
26          betas(i,j) = stats{i,j}.beta(2);
27          rsquares(i,j) = stats{i,j}.rsquare;
28      end
29  end
30  pvals = pvals';
31  end
```

Listing B.12: Matlab function for calling regstats for use in the regression analysis as described in Section 2.4.

```
1   % defmodes.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
2  %
3  % Description:  Script to generate deformation mode plots. spca ...
       struct with spca data and x struct with landmark data needs ...
       to be loaded prior to running this script. The user must ...
       choose (a) between using mean baseline or follow-up CC shape ...
       and also choose (b) whether to plot the first or last 10 ...
       stop numbers. For making choices, outcomment the lines ...
       mentioned where the text Choice (a) or Choice (b) occur.
4  %
5  % Author:       Nicolas Tiaki Otsu (s072254@student.dtu.dk)
6  % Last edited:  June 14, 2011
7  %
8  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
9  xs = [1:78 1];
10 ys = [79:156 79];
11 stop = -round(linspace(2,156,20));
12
13 % Compute mean baseline and follow-up shapes
14 x.bl_mean = mean(x.ed.bl,2);
15 x.fu_mean = mean(x.ed.fu,2);
16
17 % Choice (b): Select first/second following line to select ...
       first/last 10
18 % stop numbers
19 spca.sl = spca.sl(1:10); spca.sv = spca.sv(1:10);
20 % spca.sl = spca.sl(11:20); spca.sv = spca.sv(11:20);
21
22 c = size([spca.sl(1).k10],2);
23 r = length(spca.sl);
24
25 % Choice (a): Select first/second following line to select ...
       baseline/follow-up
26 normed = normalize(x.bl_mean);
27 % normed = normalize(x.fu_mean);
28
29 clear LMp
30 std = 1;
31 figure('Position',get(0,'ScreenSize'))
32 for j = 1:c
33     for i = 1:r
34         LMp.bl_p(:,i) = normed + std*sqrt(abs(spca.sv(j).k10(i)) ...
               *abs(spca.sl(j).k10(:,i)));
35         LMp.bl_m(:,i) = normed - std*sqrt(abs(spca.sv(j).k10(i)) ...
               *abs(spca.sl(j).k10(:,i)));
36
37         subplot(r,c,r*(i-1) + j)
38         plot(LMp.bl_p(xs,i), - LMp.bl_p(ys,i),'g-','LineWidth',2);
39
40         if j == 1
41             ylabel(['PC' num2str(i)])
42         end
43
44         hold on
45         plot(LMp.bl_m(xs,i), - LMp.bl_m(ys,i),'r-','LineWidth',2);
46         plot(normed(xs), - normed(ys),'b-','LineWidth',2)
```

```
47          set(gca,'xtick',[],'ytick',[])
48
49          if i == 1 && j == 5
50              % Choice (a): Select first/second following line to ...
                    select baseline/follow-up
51              title(['Mean baseline shape (blue), ...
                    plus/minus',num2str(std),' standard deviation ...
                    (green/red)'],'FontSize',14)
52  %             title(['Mean follow-up shape (blue), plus/minus ...
        ',num2str(std),' standard deviation (green/red)'],'FontSize',14)
53
54          end
55          axis([-.18 .19 -.09 .08])
56      end
57
58      if i == r
59          % Choice (b): Select first/second following line to ...
                select first/last 10 stop numbers
60          xlabel(['n = ' num2str(-stop(j))])
61  %         xlabel(['n = ' num2str(-stop(j+10))])
62      end
63  end
```

Listing B.13: Matlab script for computing and visualizing the deformation modes as described in Section 3.3.4.

Appendix C

Beta values and p values

	n = 2	n = 10	n = 18	n = 26	n = 34	n = 43	n = 51	n = 59	n = 67	n = 75
PC1	-1.6959	-0.9804	-0.8207	0.7883	0.7083	-0.6217	-0.5756	-0.5515	-0.5201	0.5352
PC2	-1.8258	0.3769	0.0066	-0.0552	-0.0738	-0.1046	-0.1039	-0.1250	-0.1395	-0.1621
PC3	1.0338	-1.1008	0.0889	0.2888	0.3160	0.3814	0.3696	0.3689	0.3982	-0.3341
PC4	0.3775	-0.3086	-0.0969	1.0032	0.1603	0.7753	-0.7349	-0.6206	-0.2802	-0.3477
PC5	0.7864	-0.5412	-0.9185	0.1743	0.9176	0.1139	0.2149	-0.2859	-0.6006	0.5925
PC6	2.9157	1.7972	1.2304	0.7795	-0.7346	-0.8093	-0.6163	-0.5665	-0.5210	-0.5782
PC7	2.4736	1.6799	1.3859	-1.0979	-0.7435	-0.5977	-0.7134	-0.5664	-0.3771	0.5824
PC8	3.1367	-1.3085	-0.7424	-0.8158	-0.7280	0.6773	0.7011	0.6502	0.6832	-0.2328
PC9	-1.6735	-1.4057	-1.1359	-0.6527	-0.6825	-0.6323	0.6367	0.6470	0.6022	-0.4669
PC10	1.1203	-0.9268	-0.8215	-0.7656	0.7067	-0.6576	-0.6521	-0.6294	-0.5391	-0.5192
	n = 83	n = 91	n = 99	n = 107	n = 115	n = 124	n = 132	n = 140	n = 148	n = 156
PC1	0.5073	-0.4870	-0.4776	-0.4691	-0.4654	-0.4645	-0.4588	-0.4488	-0.4273	-0.4002
PC2	-0.3063	-0.2499	-0.2024	-0.2238	-0.1243	-0.1723	-0.2768	-0.2606	-0.2065	0.5858
PC3	0.2248	0.3101	-0.3573	-0.1080	-0.0546	-0.1645	-0.1398	0.2318	0.0125	-0.5032
PC4	-0.3883	-0.3725	-0.2349	1.3840	1.5319	-0.0818	-0.7982	1.0551	0.6236	0.5142
PC5	0.5352	-0.6206	-0.6417	-0.2534	-0.5335	-0.5336	1.2418	-0.4668	0.8711	1.2800
PC6	-0.4743	-0.4683	-0.4958	-0.4815	-0.4791	-0.5783	-0.3650	0.0783	1.4445	-0.5668
PC7	-0.5318	-0.4955	0.5291	-0.6632	-0.3835	-0.4002	-0.4883	-0.3734	-1.7551	-2.2644
PC8	-0.4040	0.5275	0.6807	-0.4015	0.5511	0.5506	-0.3656	-0.5005	-0.1161	1.5326
PC9	-0.5326	-0.4781	-0.3983	0.5722	-0.2084	-0.2355	0.6071	0.6749	0.6766	10.0515
PC10	-0.5153	0.5124	0.3875	0.4045	0.3598	0.3781	0.3996	0.3782	1.5765	0.6004

Table C.1: Beta coefficients from the regression analysis for the first clinical variable (MEMORY). There has not been detected any significant corresponding p values to this clinical variable.

	n = 2	n = 10	n = 18	n = 26	n = 34	n = 43	n = 51	n = 59	n = 67	n = 75
PC1	3.5521*	1.3971	0.9303	-1.1753	-0.8047	0.9927*	0.8045	0.7534	0.6574	-0.5573
PC2	4.9892	1.1142	1.1377	1.1241	1.1155	1.1027	1.1070	1.0975	1.0869	1.0877
PC3	2.5297	1.6886	-1.0726	-0.9823	-1.0125	-0.9831	-0.9200	-0.9213	-0.7948	0.7460
PC4	1.8378	-1.7997	-1.6530	-0.2507	-1.4347	-0.3194	0.3138	0.4145	1.1973	1.0087
PC5	-3.9319*	-1.3215	1.4150	-1.5180	-0.2494	-1.5305	-1.3537	1.2710	0.4048	-0.9447
PC6	-2.1526	-3.0712**	-0.2947	-0.8894	0.4158	0.8660	1.0632	0.9907	0.9067	0.4033
PC7	1.0139	-0.5300	-2.2864*	1.8804*	1.3803*	1.1354	0.7598	0.9324	0.8999	-0.9189
PC8	-3.3599	0.9344	1.5940	1.2720	1.1769	-0.7506	-0.6364	-0.9875*	-0.9329	0.7495
PC9	2.6134	1.6255	1.3938	1.2970	1.0834	0.8838	-0.7687	-0.5721	-0.5589	0.8719
PC10	1.6710	1.8223	1.5741	1.3469	-1.0636	0.9810	0.7821	0.8242	0.7585	0.7270
	n = 83	n = 91	n = 99	n = 107	n = 115	n = 124	n = 132	n = 140	n = 148	n = 156
PC1	-0.5412	0.6096	0.5646	0.5616	0.5534	0.5497	0.5435	0.5431	0.5405	0.5425
PC2	0.9589	0.8986	0.8713	0.8484	0.8516	0.7614	0.8008	0.8007	0.8044	0.2258
PC3	-0.6988	-0.7641	-0.2462	-0.2056	0.1019	0.2481	0.2981	0.2869	-0.4620	0.3720
PC4	0.8991	0.8934	0.6598	-0.1757	-0.7324	1.7688	1.6397	-2.4429**	-3.3109**	-3.2101**
PC5	-0.8686	1.0542*	1.0291	0.7910	1.0050	1.0118	-2.4278**	1.0000	1.0137	-0.8363
PC6	1.0334	0.4325	0.4037	0.3994	0.4034	0.2815	1.2282	1.9970	-4.7236***	-5.2204**
PC7	0.4213	0.9971	-0.9043*	1.0771*	0.8164	0.7333	0.4667	0.6811	1.6872	2.4170
PC8	0.8002	-0.8692	-0.9423	0.8240	-0.8807	-0.9032	0.9732	0.4962	0.3469	2.3251
PC9	0.7377	0.7437	0.8523	-0.9478*	0.8154	0.9155	-1.1175*	-1.2994*	0.8588	4.2088
PC10	0.7174	-0.4796	-0.8049	-0.8112	-0.7455	-0.7861	-0.7984	-0.7520	0.9102	-1.5150

Table C.2: Beta coefficients from the regression analysis for the second clinical variable (SPEED). One star signifies a corresponding p value with a significance level within 10 percent. Two and three stars signify 5 and 1 percent, respectively. There are no significant p values within a 0.1 percent level.

	n = 2	n = 10	n = 18	n = 26	n = 34	n = 43	n = 51	n = 59	n = 67	n = 75
PC1	2.4961	1.2929	0.9549	-0.6343	-0.7384	0.6002	0.5063	0.5041	0.3786	-0.4699
PC2	-1.9660	-1.9488	-1.7114	-1.6945	-1.6706	-1.5633	-1.5843	-1.4024	-1.3681	-1.3152
PC3	-0.0943	0.1769	0.4187	0.2474	0.2546	0.1617	0.1555	0.1811	0.2628	-0.2894
PC4	1.5636	-0.2622	-0.1235	-0.4786	-0.0757	-0.4721	0.4635	0.5023	0.3283	0.3334
PC5	0.0852	0.6422	0.3043	-0.0867	-0.4506	-0.0750	-0.1312	0.3665	0.4849	-0.0610
PC6	-3.5927	-2.4869	-0.5390	-0.9094	-2.2665	-1.4504	0.1884	-0.9121	0.2368	0.4269
PC7	-4.5137	-0.8060	-1.7672	1.4845	1.1292	0.1943	0.7551	0.2167	-0.2040	-0.5488
PC8	-0.8255	1.5080	1.3721	0.3398	0.3330	-0.6697	-0.6663	-0.5964	-0.2695	-0.2587
PC9	2.1208	0.3899	0.5374	1.1295	0.8508	0.6866	-0.2664	-0.5872	-0.5389	0.2433
PC10	-3.5411	1.4131	1.2011	1.0122	-0.5620	0.4119	0.7534	0.2706	0.5139	0.4845
	n = 83	n = 91	n = 99	n = 107	n = 115	n = 124	n = 132	n = 140	n = 148	n = 156
PC1	-0.4418	0.4587	0.4287	0.4187	0.4135	0.4152	0.4065	0.3950	0.3536	0.2775
PC2	-0.5371	-0.5367	-0.4987	-0.4933	-0.6620	-0.5190	-0.4058	-0.5467	-0.8620	-2.0357
PC3	0.4231	0.3297	1.8513	1.3044	0.6146	0.5094	0.6687	0.4331	0.1069	-0.3216
PC4	0.4558	0.4102	-0.5301	-1.1466	-0.5917	0.6842	1.4275	0.3031	1.4941	0.2937
PC5	-0.3694	0.1292	-0.0885	-0.7906	-0.2263	-0.2541	0.1652	0.5903	-0.4288	0.4628
PC6	-0.9210	0.3323	0.3392	0.3186	0.3205	0.2674	0.5735	0.5555	-2.7058	-4.0824
PC7	0.4376	-0.7792	-0.4335	-0.0140	0.3665	0.4096	0.3112	-1.2537	0.8376	1.5182
PC8	0.0408	-0.4381	0.3287	0.4465	-0.5563	-0.5515	-1.1783	0.2835	-0.4858	-2.7803
PC9	0.2668	0.1876	0.4567	-0.5654	-1.1118	-1.2511	-0.7593	-0.8887	0.1431	0.6026
PC10	0.5123	-0.4184	0.0190	0.0151	0.0252	0.0360	0.0253	-0.0714	-0.7753	-15.7894***

Table C.3: Beta coefficients from the regression analysis for the third clinical variable (EXECUTIVE). One star signifies a corresponding p value with a significance level within 10 percent. Two and three stars signify 5 and 1 percent, respectively. There are no significant p values within a 0.1 percent level.

	n = 2	n = 10	n = 18	n = 26	n = 34	n = 43	n = 51	n = 59	n = 67	n = 75
PC1	-0.5102	1.7397	0.8212	0.5311	-0.4060	-0.6241	-0.3194	-0.2602	-0.3440	-0.0197
PC2	23.8886	11.5593	9.5214	8.6176	8.2421	7.4311	7.4205	6.5544	6.3610	6.2672
PC3	-8.3068	-2.1259	3.0232	2.7061	2.5147	2.2359	1.9865	1.8301	0.2299	-0.5538
PC4	4.3146	-1.0698	-0.3029	0.6039	0.0287	0.6809	-0.3197	-0.7622	-0.5225	-0.8153
PC5	6.1348	4.0748	-1.2702	0.4927	0.5864	-0.6663	-0.4812	-0.4511	-0.8462	1.0814
PC6	-7.1429	0.8408	1.4445	-0.7230	8.5967	6.9064	-1.3414	3.6592	-1.5352	-0.8003
PC7	23.6474	0.6209	0.7846	-0.9463	-1.0682	-1.6026	-2.2978	-1.5347	-1.4283	0.9810
PC8	0.3545	2.6117	-1.3874	-1.0316	-1.0059	-0.2992	-0.5341	0.9127	1.0603	-0.5512
PC9	4.6462	-0.9414	-0.5512	-1.3007	-0.3133	-0.2442	0.5317	-0.4002	-0.2578	-1.5678
PC10	-1.8310	0.2401	-0.0669	-0.0581	0.5687	-0.6627	0.1114	-0.6871	-0.3271	-0.3391
	n = 83	n = 91	n = 99	n = 107	n = 115	n = 124	n = 132	n = 140	n = 148	n = 156
PC1	0.0761	-0.2107	-0.1579	-0.1680	-0.1677	-0.1967	-0.2448	-0.2641	-0.2596	-0.1365
PC2	2.3191	2.2395	2.0295	2.0772	3.1320	2.1917	1.6824	2.2476	3.3011	4.4120
PC3	0.4488	0.7249	-9.4947*	-10.3506*	-9.2978	-8.1433	-8.3864	-9.3334	5.1890	10.1050*
PC4	-0.9878	-1.1205	1.0804	-11.2484	-7.4997	-9.8133	-9.0399	-1.0162	1.0696	3.5206
PC5	1.0897	-0.4039	0.0278	2.8762	0.9791	1.2823	-1.8169	-6.6514	2.4320	-1.6763
PC6	6.3370	-0.8957	-0.6845	-0.5572	-0.6331	-0.6037	-1.0859	-0.0002	-1.0851	-5.1258
PC7	-0.8761	5.7443	0.8100	0.3877	-1.0018	-1.0686	-0.8933	9.0594	-1.5553	7.6556
PC8	-1.3899	1.1136	-4.8040	-0.9364	1.0550	1.0216	5.9919	-1.1438	9.4573	-0.6687
PC9	-0.9267	-1.3002	-0.8522	1.0363	4.2616	5.8998	0.6012	0.6525	-10.9479	-1.8013
PC10	-0.2799	0.0492	1.0505	0.9244	0.7884	0.8980	0.8159	2.1642	-5.7332	-13.8643

Table C.4: Beta coefficients from the regression analysis for the fourth clinical variable (verbal). One star signifies a corresponding p value with a significance level within 10 percent. Two and three stars signify 5 and 1 percent, respectively. There are no significant p values within a 0.1 percent level.

	n = 2	n = 10	n = 18	n = 26	n = 34	n = 43	n = 51	n = 59	n = 67	n = 75
PC1	-2.9865	-0.7150	-0.6600	1.0917	0.6491	-0.9820	-0.7657	-0.7122	-0.5983	0.4571
PC2	9.6370	5.2184	4.0466	3.5304	3.3547	3.0160	2.9518	2.6494	2.5347	2.4571
PC3	-7.2650	-1.8257	2.8621	2.5579	2.3893	2.0644	1.8667	1.7401	1.7661	-1.6121
PC4	4.7820	-1.4048	-0.6971	0.2416	-0.3236	0.3403	-0.2172	-0.4851	-0.2555	-0.4873
PC5	5.1491	3.5250	-1.4339	0.0578	0.2299	-0.7904	-0.5562	-0.1511	-0.4975	1.1057
PC6	1.1714	2.8944	0.5323	0.6604	-1.5624	-1.8704	-1.0457	-1.9993	-1.0141	-0.4673
PC7	11.0795	0.6661	2.1203	-1.8681	-1.4805	-1.1385	-1.3727	-1.0455	-1.4212	1.0965
PC8	2.7224	-0.6265	-1.8514	-1.2630	-1.1659	0.6286	0.4666	1.0781	1.0502	-0.6869
PC9	-0.4930	-1.4207	-1.2563	-1.5021	-1.0078	-0.8283	0.6575	0.4055	0.4155	-0.9867
PC10	-2.7179	-1.4104	-1.3177	-1.1314	1.0090	-0.9503	-0.7042	-0.7550	-0.7335	-0.6987
	n = 83	n = 91	n = 99	n = 107	n = 115	n = 124	n = 132	n = 140	n = 148	n = 156
PC1	0.4586	-0.5627	-0.5003	-0.5028	-0.4949	-0.5045	-0.5178	-0.5147	-0.5052	-0.4537
PC2	0.7659	0.7687	0.7280	0.6942	1.2187	0.6856	0.4273	0.5584	0.8797	1.8406
PC3	1.3401	1.4286	-4.1620*	-4.4924*	-4.2344*	-3.9661	-3.9220	-3.7016	2.7299	4.0116
PC4	-0.6561	-0.7581	-0.6912	-0.6419	0.5226	-1.1254	-2.8960	1.3498	2.3788	1.7555
PC5	1.1398	-0.8187	-0.7172	-0.0088	-0.1329	-0.0216	1.0383	-1.3736	3.6268	5.6909
PC6	1.4334	-0.6067	-0.4583	-0.4655	-0.5102	-0.6016	-0.7887	0.4947	4.2946	3.1056
PC7	-0.5503	1.2934	1.0552	-0.5734	-0.7183	-0.6594	-0.6031	1.5891	-1.4834	0.5744
PC8	-1.1201	1.1780	-0.6046	-0.7496	1.1441	1.1689	0.9914	-0.8467	-0.8149	-7.2125
PC9	-0.8401	-0.9279	-0.6967	1.2124	0.5625	1.0800	1.1235	1.4038	-0.2676	0.1869
PC10	-0.6868	0.3864	1.0108	1.0214	0.9623	1.0791	1.1252	1.7941	0.8667	1.8506

Table C.5: Beta coefficients from the regression analysis for the fifth clinical variable (gdstotal). One star signifies a corresponding p value with a significance level within 10 percent. Two and three stars signify 5 and 1 percent, respectively. There are no significant p values within a 0.1 percent level.

	n = 2	n = 10	n = 18	n = 26	n = 34	n = 43	n = 51	n = 59	n = 67	n = 75
PC1	0.6042	0.5729	0.5263	0.5083	0.5088	0.5258	0.5143	0.5181	0.5297	0.5103
PC2	0.7150	0.8765	0.9974	0.9766	0.9681	0.9527	0.9527	0.9400	0.9316	0.9193
PC3	0.8500	0.5348	0.9678	0.8795	0.8585	0.8053	0.7990	0.7936	0.7893	0.8149
PC4	0.9406	0.9031	0.9646	0.5005	0.9328	0.5296	0.5400	0.5605	0.8591	0.7952
PC5	0.8389	0.8600	0.5267	0.9240	0.5092	0.9565	0.9086	0.8617	0.5770	0.5844
PC6	0.4514	0.4644	0.4896	0.5236	0.8001	0.7616	0.6232	0.7925	0.6554	0.5795
PC7	0.6351	0.4536	0.4697	0.5000	0.5584	0.6624	0.5778	0.6349	0.7715	0.5274
PC8	0.4442	0.4714	0.6476	0.5316	0.5498	0.5015	0.4977	0.5064	0.5124	0.8539
PC9	0.6320	0.4436	0.4536	0.6450	0.5771	0.5497	0.5181	0.5023	0.5051	0.6832
PC10	0.8456	0.6099	0.6024	0.5846	0.5132	0.5258	0.5412	0.5266	0.5169	0.5230
	n = 83	n = 91	n = 99	n = 107	n = 115	n = 124	n = 132	n = 140	n = 148	n = 156
PC1	0.5146	0.5181	0.5184	0.5191	0.5209	0.5199	0.5197	0.5245	0.5355	0.5551
PC2	0.7959	0.8328	0.8670	0.8487	0.9251	0.8852	0.7963	0.8161	0.8690	0.7091
PC3	0.8783	0.8169	0.8408	0.9540	0.9765	0.9258	0.9372	0.9045	0.9940	0.7970
PC4	0.7452	0.7468	0.8610	0.6048	0.4858	0.9745	0.7277	0.5425	0.8003	0.8470
PC5	0.5599	0.5525	0.5485	0.8523	0.6116	0.6135	0.4784	0.8189	0.7257	0.6548
PC6	0.7544	0.6398	0.6167	0.6142	0.6233	0.6174	0.7919	0.9705	0.5696	0.8777
PC7	0.6062	0.7315	0.5509	0.5254	0.7014	0.6982	0.6535	0.8478	0.5043	0.6042
PC8	0.7154	0.5693	0.6273	0.6770	0.5346	0.5409	0.8148	0.6721	0.9671	0.7576
PC9	0.5433	0.6198	0.6909	0.5411	0.8863	0.8801	0.5424	0.5441	0.8103	0.1180
PC10	0.5191	0.5183	0.7089	0.6902	0.7233	0.7147	0.6994	0.7556	0.5591	0.9308

Table C.6: p values from the regression analysis for the first clinical variable (MEMORY).

	n = 2	n = 10	n = 18	n = 26	n = 34	n = 43	n = 51	n = 59	n = 67	n = 75
PC1	0.0765	0.1907	0.2423	0.1079	0.2217	0.0986	0.1375	0.1504	0.1959	0.2645
PC2	0.1037	0.4549	0.3563	0.3312	0.3247	0.3080	0.3034	0.2814	0.2765	0.2680
PC3	0.4517	0.1207	0.4281	0.4012	0.3524	0.3010	0.3021	0.2876	0.3855	0.3950
PC4	0.5555	0.2476	0.2171	0.7844	0.2194	0.6738	0.6706	0.5273	0.2166	0.2200
PC5	0.0973	0.4835	0.1119	0.1759	0.7706	0.2328	0.2390	0.2071	0.5411	0.1554
PC6	0.3659	0.0412	0.7880	0.2362	0.8157	0.5975	0.1673	0.4544	0.2060	0.5298
PC7	0.7519	0.7008	0.0516	0.0594	0.0763	0.1766	0.3348	0.2031	0.2592	0.1040
PC8	0.1821	0.4029	0.1095	0.1119	0.1151	0.2253	0.3166	0.0999	0.1451	0.3345
PC9	0.2234	0.1490	0.1341	0.1356	0.1492	0.1733	0.2039	0.3345	0.3143	0.2146
PC10	0.6366	0.1019	0.1037	0.1169	0.1088	0.1230	0.2330	0.1769	0.1373	0.1450
	n = 83	n = 91	n = 99	n = 107	n = 115	n = 124	n = 132	n = 140	n = 148	n = 156
PC1	0.2578	0.1877	0.2139	0.2090	0.2139	0.2151	0.2143	0.2099	0.2020	0.1927
PC2	0.1871	0.2163	0.2404	0.2389	0.2943	0.2989	0.2238	0.2447	0.2958	0.8152
PC3	0.4387	0.3532	0.8220	0.8582	0.9286	0.8194	0.7848	0.8092	0.6494	0.7572
PC4	0.2203	0.2072	0.4237	0.9150	0.5881	0.2607	0.2442	0.0211	0.0280	0.0491
PC5	0.1230	0.0999	0.1168	0.3443	0.1190	0.1185	0.0235	0.4248	0.5067	0.6349
PC6	0.2672	0.4821	0.5075	0.4966	0.5012	0.6927	0.1479	0.1244	0.0022	0.0203
PC7	0.5066	0.2609	0.0966	0.0927	0.1836	0.2474	0.4854	0.5692	0.2963	0.3680
PC8	0.2397	0.1265	0.2740	0.1637	0.1057	0.1019	0.3103	0.4950	0.8410	0.4462
PC9	0.1703	0.2088	0.1657	0.0989	0.3626	0.3401	0.0673	0.0566	0.6203	0.2881
PC10	0.1436	0.3254	0.2066	0.1929	0.2324	0.2158	0.2089	0.3138	0.5835	0.7216

Table C.7: p values from the regression analysis for the second clinical variable (SPEED).

	n = 2	n = 10	n = 18	n = 26	n = 34	n = 43	n = 51	n = 59	n = 67	n = 75
PC1	0.3731	0.3856	0.3896	0.5348	0.4216	0.4749	0.5035	0.4908	0.5937	0.5001
PC2	0.6465	0.3481	0.3194	0.2932	0.2899	0.2998	0.2906	0.3235	0.3258	0.3367
PC3	0.9840	0.9074	0.8245	0.8796	0.8669	0.9030	0.9005	0.8809	0.8370	0.8130
PC4	0.7190	0.9039	0.9473	0.7078	0.9630	0.6553	0.6522	0.5826	0.8083	0.7715
PC5	0.9795	0.8071	0.8068	0.9559	0.7054	0.9666	0.9349	0.7944	0.5995	0.9477
PC6	0.2787	0.2372	0.7242	0.3853	0.3617	0.5258	0.8610	0.6213	0.8130	0.6333
PC7	0.3120	0.6751	0.2819	0.2870	0.2994	0.8685	0.4919	0.8322	0.8547	0.4872
PC8	0.8144	0.3327	0.3240	0.7613	0.7498	0.4381	0.4521	0.4771	0.7632	0.8113
PC9	0.4789	0.8044	0.6793	0.3520	0.4172	0.4485	0.7525	0.4775	0.4866	0.8041
PC10	0.4724	0.3639	0.3741	0.3989	0.5442	0.6431	0.4102	0.7509	0.4711	0.4868
	n = 83	n = 91	n = 99	n = 107	n = 115	n = 124	n = 132	n = 140	n = 148	n = 156
PC1	0.5080	0.4776	0.4989	0.5021	0.5058	0.5023	0.5058	0.5136	0.5498	0.6332
PC2	0.5967	0.5968	0.6302	0.6237	0.5590	0.6117	0.6588	0.5693	0.4217	0.1295
PC3	0.7367	0.7741	0.2240	0.4156	0.6982	0.7366	0.6603	0.7937	0.9399	0.8479
PC4	0.6563	0.6784	0.6449	0.6171	0.7536	0.7553	0.4675	0.8384	0.4793	0.8977
PC5	0.6389	0.8853	0.9231	0.4979	0.8017	0.7791	0.9125	0.7355	0.8403	0.8505
PC6	0.4784	0.6985	0.6896	0.6973	0.7015	0.7877	0.6286	0.7598	0.2134	0.1950
PC7	0.6208	0.5289	0.5688	0.9875	0.6691	0.6434	0.7386	0.4520	0.7102	0.6852
PC8	0.9657	0.5815	0.7845	0.5889	0.4645	0.4748	0.3783	0.7798	0.8402	0.5135
PC9	0.7225	0.8204	0.5947	0.4812	0.3732	0.3496	0.3739	0.3512	0.9528	0.9132
PC10	0.4546	0.5384	0.9830	0.9861	0.9770	0.9676	0.9773	0.9453	0.7376	0.0072

Table C.8: p values from the regression analysis for the third clinical variable (EXECUTIVE).

	n = 2	n = 10	n = 18	n = 26	n = 34	n = 43	n = 51	n = 59	n = 67	n = 75
PC1	0.9601	0.7482	0.8388	0.8863	0.9033	0.8381	0.9076	0.9221	0.8939	0.9938
PC2	0.1241	0.1249	0.1267	0.1407	0.1502	0.1745	0.1727	0.2038	0.2083	0.2074
PC3	0.6257	0.7005	0.6593	0.6481	0.6486	0.6426	0.6603	0.6769	0.9605	0.9009
PC4	0.7847	0.8922	0.9644	0.8965	0.9961	0.8594	0.9318	0.8186	0.9153	0.8451
PC5	0.6105	0.6698	0.7788	0.9310	0.8923	0.9185	0.9342	0.9297	0.8009	0.7487
PC6	0.5538	0.9126	0.7947	0.8495	0.3409	0.4056	0.7314	0.5854	0.6730	0.8056
PC7	0.1444	0.9292	0.8955	0.8520	0.7873	0.7071	0.5649	0.6796	0.7240	0.7326
PC8	0.9779	0.6446	0.7839	0.7996	0.7909	0.9241	0.8684	0.7647	0.7443	0.8887
PC9	0.6695	0.8693	0.9071	0.7682	0.9345	0.9409	0.8625	0.8941	0.9271	0.6600
PC10	0.9186	0.9662	0.9891	0.9894	0.8660	0.8375	0.9733	0.8245	0.8996	0.8935

	n = 83	n = 91	n = 99	n = 107	n = 115	n = 124	n = 132	n = 140	n = 148	n = 156
PC1	0.9750	0.9285	0.9454	0.9410	0.9408	0.9304	0.9122	0.9044	0.9039	0.9485
PC2	0.5294	0.5435	0.5897	0.5696	0.4466	0.5552	0.6144	0.5196	0.3970	0.3667
PC3	0.9218	0.8621	0.0855	0.0745	0.1054	0.1380	0.1283	0.1198	0.3129	0.0962
PC4	0.7907	0.7552	0.7960	0.1762	0.2730	0.2178	0.2048	0.8507	0.8892	0.6715
PC5	0.7032	0.9013	0.9933	0.4974	0.7649	0.6968	0.7393	0.2942	0.7532	0.8509
PC6	0.1788	0.7739	0.8245	0.8515	0.8350	0.8671	0.8010	1.0000	0.8910	0.6549
PC7	0.7852	0.2007	0.7695	0.9052	0.7478	0.7397	0.7921	0.1339	0.8494	0.5737
PC8	0.6872	0.6998	0.2708	0.7552	0.7028	0.7157	0.2172	0.7562	0.2794	0.9655
PC9	0.7343	0.6649	0.7848	0.7225	0.3474	0.2245	0.8465	0.8507	0.2116	0.9286
PC10	0.9105	0.9841	0.7453	0.7700	0.8034	0.7805	0.8002	0.5674	0.4951	0.5194

Table C.9: p values from the regression analysis for the fourth clinical variable (verbal).

	n = 2	n = 10	n = 18	n = 26	n = 34	n = 43	n = 51	n = 59	n = 67	n = 75
PC1	0.5124	0.7681	0.7148	0.5111	0.6641	0.4721	0.5338	0.5494	0.6041	0.6868
PC2	0.1658	0.1216	0.1471	0.1776	0.1909	0.2183	0.2257	0.2512	0.2627	0.2695
PC3	0.3400	0.4602	0.3506	0.3347	0.3328	0.3380	0.3557	0.3757	0.3949	0.4173
PC4	0.4985	0.6909	0.8186	0.9074	0.9029	0.8432	0.8967	0.7442	0.9076	0.7941
PC5	0.3390	0.4094	0.4783	0.9819	0.9056	0.7861	0.8312	0.9473	0.7404	0.4639
PC6	0.8283	0.3979	0.8303	0.6984	0.6992	0.6149	0.5497	0.5053	0.5331	0.7481
PC7	0.1264	0.8313	0.4275	0.4102	0.4030	0.5507	0.4422	0.5295	0.4321	0.3931
PC8	0.6340	0.8047	0.4132	0.4872	0.4920	0.6546	0.7462	0.4292	0.4701	0.6966
PC9	0.9194	0.5787	0.5523	0.4467	0.5547	0.5741	0.6321	0.7630	0.7416	0.5360
PC10	0.7346	0.5775	0.5489	0.5621	0.5031	0.5108	0.6361	0.5859	0.5270	0.5374
	n = 83	n = 91	n = 99	n = 107	n = 115	n = 124	n = 132	n = 140	n = 148	n = 156
PC1	0.6727	0.5922	0.6275	0.6202	0.6244	0.6162	0.6023	0.6006	0.5993	0.6314
PC2	0.6426	0.6413	0.6656	0.6712	0.5081	0.6801	0.7750	0.7208	0.6142	0.4001
PC3	0.5124	0.4439	0.0922	0.0838	0.0993	0.1062	0.1119	0.1683	0.2353	0.1403
PC4	0.6936	0.6374	0.7117	0.8633	0.8646	0.7525	0.3645	0.5763	0.4885	0.6365
PC5	0.3728	0.5740	0.6305	0.9963	0.9277	0.9883	0.6708	0.6287	0.2941	0.1529
PC6	0.4975	0.6636	0.7400	0.7267	0.7075	0.7094	0.6825	0.8670	0.2246	0.5450
PC7	0.7020	0.5203	0.3934	0.6938	0.6064	0.6467	0.6909	0.5578	0.6857	0.9249
PC8	0.4681	0.3617	0.7571	0.5768	0.3548	0.3514	0.6487	0.6075	0.8352	0.2968
PC9	0.4916	0.4896	0.6178	0.3527	0.7819	0.6198	0.4185	0.3652	0.9457	0.9834
PC10	0.5377	0.7269	0.4847	0.4701	0.4969	0.4539	0.4351	0.2889	0.8178	0.8477

Table C.10: p values from the regression analysis for the fifth clinical variable (gdstotal).

Bibliography

[1] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on DOI - 10.1109/34.927467*, 23(6):681–685, 2001.

[2] H. Jokinen, C. Ryberg, H. Kalska, R. Ylikoski, E. Rostrup, M. B. Stegmann, G. Waldemar, S. Madureira, J. M. Ferro, E. C. W. van Straaten, P. Scheltens, F. Barkhof, F. Fazekas, R. Schmidt, L. Pantoni, D. Inzitari, and T. Erkinjuntti. Corpus callosum atrophy is associated with mental slowing and executive deficits in subjects with age-related white matter hyperintensities.the LADIS study. *Journal of Neurology Neurosurgery and Psychiatry*, 2007.

[3] C. Ryberg, E. Rostrup, M.B. Stegmann, F. Barkhof, P. Scheltens, E.C.W. van Straaten, F. Fazekas, R. Schmidt, J.M. Ferro, H. Baezner, T. Erkinjuntti, H. Jokinen, L.-O. Wahlund, J. O'Brien, A.M. Basile, L. Pantoni, D. Inzitari, and G. Waldemar. Clinical significance of corpus callosum atrophy in a mixed elderly population. *Neurobiology of Aging*, 28(6):955–963, 2007. cited By (since 1996) 6.

[4] K. Sjöstrand. *Regularized Statistical Analysis of Anatomy*. PhD thesis, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, 2007. Supervised by Assoc. Prof. Rasmus Larsen, IMM, DTU, and partly supervised by Dr. Colin Studholme (UCSF).

[5] K. Sjöstrand, E. Rostrup, C. Ryberg, R. Larsen, C. Studholme, H. Baezner, J. Ferro, F. Fazekas, L. Pantoni, D. Inzitari, and G. Waldemar. Sparse decomposition and modeling of anatomical shape variation. *Medical Imaging, IEEE Transactions on*, 26(12):1625 –1635, December 2007.

[6] Karl Sjöstrand, Mikkel B. Stegmann, and Rasmus Larsen. Sparse principal component analysis in medical shape modeling.

[7] M.B. Stegmann, B.K. Ersbøll, and R. Larsen. Fame-a flexible appearance modeling environment. *Medical Imaging, IEEE Transactions on DOI - 10.1109/TMI.2003.817780*, 22(10):1319–1331, 2003.

[8] Hui Zou, Trevor Hastie, and Robert Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286, 2006.