Bachelor’s/Master's thesis

Information and Communications Technology

2019

Rasha ElMansuri

EVALUATION OF MATLAB OPEN SOURCE SOFTWARE FOR IMAGE RECONSTRUCTION IN PET-OMEGA

TURKU AMK

TURKU UNIVERSITY OF APPLIED SCIENCES

BACHELOR THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information and Communications Technology

2019 | 46

Rasha ElMansuri

EVALUATION OF MATLAB OPEN SOURCE SOFTWARE FOR IMAGE RECONSTRUCTION IN PET – OMEGA

Abstract

Open source software is growing, and widely used in different fields of science and technology. In medical imaging and diagnostics, reconstruction software is integrated with the existing PET/CT systems, and these systems cannot be easily modified, investigated nor applied. Therefore, providing an alternative, open source imaging software in creation of PET images from any available data is taken into consideration, that is by introducing OMEGA as scientific image reconstruction software, using mainly MATLAB software, due to its popularity among the scientific users.

The main objective of this thesis is to provide initial and comprehensive evaluation, from a user-case perspective on OMEGA software as a scientific image reconstruction software for medical analysis. The thesis was accomplished in collaboration with Turku PET Center. The evaluation held throughout this thesis is achieved by testing the software algorithms for functionality and addressing possible errors. There were fifteen algorithms in total. The reconstructed images were inspected visually and by volume of Interest analysis and profile analysis to evaluate their quality.

The findings show that six algorithms out of fifteen reconstructed images, however eight algorithms were functioning and computing successfully. The analysis results preview the image quality and the scale in which images differ from each other based on each algorithm. These results highlight the importance of software evaluation, which insures usability of OMEGA software as a reconstruction software and its capabilities.

KEYWORDS:

Open Source, Medical Imaging, Positron Emission Tomography, Diagnostic Radiology, Software Evaluation, Data Analysis, Software Testing.

OPINNÄYTETYÖ AMK | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tietotekniikka ja viestintäkouluts

2019 | 46

Rasha ElMansuri

MATLAB-POHJAISEN AVOIMEN LÄHDEKOODIN OHJELMISTON ARVIOINTI KUVIEN REKONSTRUOINTIIN PET:SSÄ -OMEGA

Tiivistelmä

Avoimen lähdekoodin ohjelmiston määrä ja käyttö on kasvussa eri tieteen ja teknologian aloilla. Lääketieteellisessä kuvantamisessa ja diagnostiikkassa rekonstruktio—ohjelmisto on integroitu PET/CT-järjestelmien kanssa, eikä näitä järjestelmiä voida helposti muuttaa. Vaihtoehtoinen, avoimen lähde koodin ohjelmisto PET-kuvien rekonstruktioon - OMEGA voisi olla hyödyllinen useille tieteelliselle käyttäjille.

Tämän tutkielman päätavoitteena on antaa alustava ja kattava arviointi OMEGA-ohjelmiston käytöstä PET-kuvien rekonstruoitiin. Tutkielma toteutettiin yhteistyössä Valtakunnallisen PET-keskuksen kanssa. Ohjelmistossa testattiin: algoritmien toiminnallisuutta, mahdollisia virheitä. Yhteensä algoritmeja testattiin viisitoista. Rekonstruoituja kuvia analysoitiin VOI-analyysin ja profiili analyysin avulla ja niiden laatua arviointiin.

Havainnot osoittavat, että kuusi algoritmia viidestätoista tuottivat rekonstruoidun kuvan. Analyysissä arvoitiin kuvan laatua jotka osoittivat algoritmien välisiä kuvanlaadullisia eroja. Nämä tulokset korostavat ohjelmiston arvioinnin merkitystä, joka varmistaa OMEGA-ohjelmiston käytettävyyden.

ASIASANAT:

Avoin Lähdekoodi, Lääketieteellinen Kuvantaminen, Positroniemissiotomografia, Diagnostinen Radiologia, Ohjelmiston Arviointi, Tietojen Analysointi, Ohjelmistojen Testaus.

CONTENTS

FIGURES

PICTURES

TABLES

ACKNOWLEDGEMENT

This work would have not been achieved and completed without the grace of God and all the golden-hearts that kindly helped me throughout this process, and shared with me their valuable experiences, both academically as well as personally. This journey began from my loving and supporting Libyan household. My mother Naziha Mansuri, my father Mansur ElMansuri and my two beloved brother Rami and Ramez ElMansuri; my beloved family, you have always been there for me, giving me all the support I need and unconditionallylove me. I love you and I am forever grateful for you.

Working in Turku PET Centre as a trainee, while pursuing my bachelor’s degree in Turku University of Applied Sciences in the field of ICT engineering, helped me recognize different parts of academic interests and how information technology, or technology in general, is a crucial part of different scientific fields; especially medicine and more specifically medical imaging. Pursuing my personal interest, this thesis is conducted in a way that combines both fields of software development and medical imaging together, focusing on image reconstruction.

To my respectful thesis supervisors; I would like to express my gratitude and appreciation, for making this thesis possible, by granting me the tools needed for conducting this research. I would like to thank Chunlei Han for providing me the chance to return to Turku PET Centre and introducing me to my second supervisor, Jarmo Teuho; that I would like to thank, for providing me his valuable time and continuous support, as well as motivating me during each phase throughout conducting this thesis. I would also like to thank the OMEGA team in Kuopio for allowing this experience.

To my respectful instructor in Turku University of Applied Sciences, Jari-Pekka Paalassalo. I would like to express my graditude and thank JP, for his flexibility by allowing me to express my academic interest, in the field of software development and its relation to the field of medical imaging and radiology. Also, to my respectful teachers and instructors at Turku University of Applied Sciences; Mauri Suhonen, Sanna Määttä, Poppy Skarli, Hazem Al-Bermanei, Tiina Ferm, Tapani Ojenperä, Timo Tolmunen, along with Päivi Oliva, Elina Karaus, Janni Ahtinen, and all the TUAS staff and teachers, that have helped me in anyway throughout my academic years; you have been such great source of support and strongly guided me professionally and academically throughout this chapter of my life. I appreciate you.

LIST OF ABBREVIATIONS

2-D	Two-Dimensional
3-D	Three-Dimensional
4-D	Four-Dimensional
CCP	Collaborative Computational Project
COSEM	Complete-data Ordered Subsets Expectation Maximization
CT	Computed Tomography
DQE	Detective Quantum Efficiency
EPSRC	Engineering and Physical Sciences Research Council
EIBIR	European Institute for Biomedical Imaging Research
FWHM	Full Width at Half Maximum
GPL	General Public License
MRI	Magnetic Resonance Imaging
OMEGA	Open-Source MATLAB Emission Tomography Software
NIFTI	Neuroimaging Informatics Technology Initiative
OS	Operating System
OSS	Open Source Software
PACS	Picture Archiving and Communication System
PC	Personal Computer
PET	Positron Emission Tomography
PSF	Point Spread Function
RDP	Windows Remote desktop connection

ROI	Region of Interest
SD	Standard Deviation
SF	Screen-Film
SNR	Signal-to-noise ratio
PECT	Single Photon Emission Computed Tomography
STIR	Software for Tomographic Image Reconstruction
TYKS	Turun Yliopistollinen Keskussairaala – Turku University Hospital
UI	User Interface
UX	User Experience
VM	Virtual Machine
VOI	Volume of Interest
WS	Wiener Spectra

1. INTRODUCTION

Imaging systems used in nuclear medicine, relay on technologies for creating a visual interpretation of raw data collected from an object under medical analysis intervention. Due to the constant evolution of technology, and thanks to the solid ground they maintained, techniques such as PET/CT scanners, and systems are improving, in physical performance and image quality; furthermore, such as the importance of technology, software is a crucial part for completing the puzzle.

Software is the key component in creation of both quantitatively as well as qualitatively accurate medical images. The purpose of this thesis, is to study the usage of scientific and open source software in the medical imaging field, focusing mainly on evaluation of software usability, by assessing the ease of application of an existing software, as well as testing and performing data analysis, ensuring the quality of the images resulted.

- Thesis Background

This thesis is conducted in collaboration with Turku PET Centre. Turku PET Centre was initiated in the 1970, with the inceptive aim is to develop target for isotope production. The key purposes of the PET Centre are mainly based on the compatibility with different universities around the city of Turku, including, for example University of Turku and Åbo Academy University.

The facilities of the PET Centre are allocated in university campus and image clinical laboratories. During 1996, Turku PET Centre was granted national status from the Ministry of education. Due to this establishment, increase in funding came along, which increased the research activities and closer collaborations started establishing with laboratories around cities and countries within EU, USA, and Asia. Turku PET Centre’s scientific research department involves cardiovascular and metabolic research, cancer research, radiochemistry, neurology and more research departments.

The amount of imaging studies is around 5800 study annually and nearly 2000 high quality peer-reviewed scientific publications. The imaging studies and research activities in Turku PET Centre are mainly conducted using PET scan in medical imaging, as well as ultrasound, MRI and CT scan (Turkupetcentre.fi, 2019).

The different technology and techniques used in Turku PET Centre are relying on computers and computer software, for performing nearly all the data analysis, statistics and for data storage. Besides the physics and engineering technology of each scanner, such as PET and CT, Turku PET Centre is interacting with imaging software at a high level.

Software in the Turku PET Centre is used by scientists, software engineers, doctors as well as researchers for their research purpose; however, most of the reconstruction software used is implemented on existing PET/CT systems, which makes the software difficult to be modified, investigated nor applied. Thus, it would be of interest to apply an alternative, open source imaging software in creation of PET images from the available data. Furthermore, before the process of providing an alternative software solution, the existing software platforms need to be investigated in terms of application, availability, and ease of use and installation process.

This thesis covers the detailed set-by-step installation, testing data, and overall evaluation of the alternative and open source software OMEGA v.0. The technology stack includes MATLAB v.R2017a, which is a framework and a software developed by Math Works, where testing and running the software OMEGA v.0 is mainly done. The image analysis software stack includes Carimas v.2.10, which is a software developed by the PET Centre for processing images and analysis, along with software Vinci v.4.91. The evaluation and installation of software OMEGAv.0, is taking place on a remote Linux machine.

This thesis starts with background nature, including a brief background knowledge of PET, and image reconstruction and its key concepts, to provide better understanding to the reader. Detailed information includes the technologies used, step-by-step guide to installation, testing and evaluation from a none scientific user-case perspective will be discussed in upcoming chapters.

2. THEORETICAL BACKGROUND

2.1. Computers in Nuclear Medicine

The enormous growth of computer potentials increases their usability in the diagnostic radiology fields of medicine. Computers are needed for performing CT scan, MRI scan, microprocessor-controlled X-ray generators, ultrasound applications, as well as quality assurance and system control (Dowsett, Kenny and Johnston, 2006). The information and the data are taken into computer systems to be stored and compared, then reports can be generated. Medical imaging generators, such as diagnostic radiology, pathology laboratories and cardiology, need image storage space and transmission in order to keep track of the clients; therefore, computers are used to transfer data and archive or retrieve old data, as well as providing the memory needed for such performances, bettering future research and assuring its quality.

Furthermore, computers provide not only the necessary memory and processors, but also the platforms for large collections of different application software's. Due to the computer's graphical and advanced user interfaces, imaging diagnostics techniques can be performed. Computers are considered a great tool for clinical radiology, for example, applications for image communicating and management, laboratory and radio-pharmacy management, as well as the programs for performing office tasks, such as Microsoft word and Latex for writing reports and different documents.

Besides the collection of computer programs that can be used offline, a considerable number of programs can be easily accessible throughout the internet, such as open sources application software and license software, enhancing the computer's experience and making the computer a critical and a beneficial tool in clinical medicine (D. Margen and S. Candrlic, IEEE, 2015). In medical imaging, and especially when interacting with PET and CT scanners, computers are important; not only for accurate measurements tracking and providing storage for saving these measurements, yet also computers are important for complicated tasks performances (Bailey, Humm, Todd-Pokropek and Aswegen, 2014). Thus, without the help of the computer technology, it would not be possible to execute images nor results of any sort.

The main software evaluated throughout this thesis, OMEGA v.0 is still under development (Villekf/OMEGA, GitHub, 2019). The version used while conducting this thesis is OMEGA v.0. Nevertheless, OMEGA is considered an OS software using the GPL license, insuring that it is a copy-left software that can be publicly used freely, allowing contributions to the code from a large audience. The difference between free software and software freedom needs is addressed as following; free software is related to materialistic value, for example the software price, however, the GPL license insures freedom of software, meaning providing the ability to run, copy, change distribute and further development the software by its users (Gnu.org, 2019).

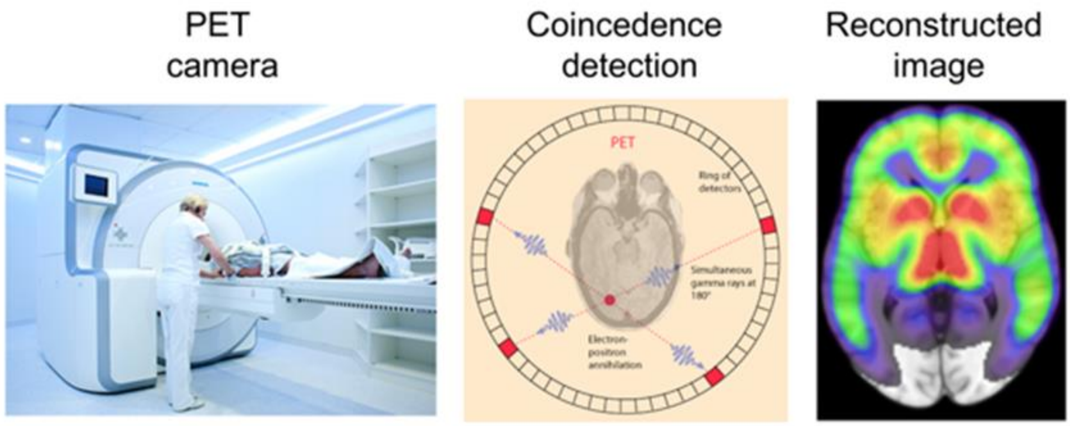
2.2. Basics of PET

According to James A. Sorenson, author of Physics in Nuclear Medicine book (2013); Positron Emission Tomography (PET) is considered the second largest method used for tomographic imaging in nuclear research. Tomography is a noun, with the Greek origin of Tomos, meaning section or slice. The technique used in radiology for imaging by cross section through a body (of animal or human) or an object, using ultrasounds or X-Rays is called Tomography (Basics PET Physics and Instrumentation, Humanhealth.iaea.org, 2016).

PET is the technique known for using the least amount of radioactive materials named radiotracers, or radioactive tracers; which are chemical compounds, where one or more atoms are replaced by a radioactive Isotope. In The physics of Diagnostic Imaging book, PET scan detector uses a special camera and a computer to evaluate the tissue and organ function of a living system or phantom to produce a tomographic image based on the radiotracer distribution.

The PET scanner detects the “back-to -back” of photons being produced when a positron interacts with an ordinary electron. PET has granted clinical acceptance and is widely spreading; nowadays, PET stands firmly alongside planar 2-D imaging and PET in clinical nuclear medicine; because of its ability to detect diseases on an early stage compared to other imaging tests, as it is used commonly to detect tumor cells, such as cancer cells, since they are metabolically active, heart problems and nervous system related diseases (Dowsett D., Kenny and P.R Johnston2006).

The main differences between PET test and other imaging tests, is its ability to precisely reveal metabolic changes in tissues or organs at an anatomical level. Picture1. shows one of the main advantages of PET scanner, which is the in-vivo measurements of different function processing, for instance, perfusion and metabolism; also, PET test can allocate and measure the distribution of neuro receptors, by detecting the subnanmolar concentration of labeled drugs (Turkupetcentre.fi, 2019).



Picture 1. PET system and coinicidence detection process with an example of resulting image.

2.3. Basics of PET Image Reconstruction

2.3.1. Reconstruction Method

Reconstruction is basically the procedure where PET images are being formed from the raw data collected during the acquisition of PET. The next subchapter discusses the Sinogram in more details, explaining its formation and definition (M. S., and Qi, J. Iterative image reconstruction for PET, Physics in medicine and biology, 2009). PET and other tomography scanners such as CT use detector systems surrounding the body being imaged. These detectors capture the photons emitted from the body from different angular views; this different angular view is known as projections.

As discussed in the previous chapter, PET has very sensitive detectors capabilities. PET scanners can acquire projection data, which is then formed to 3-D images with the application of a 3-D reconstruction algorithm. Multiple algorithms have been developed based on iterative reconstruction methods. According to Quantitative Analysis in Nuclear Medicine Imaging, edited by Habib Zaidi, and published on Elsevier USA (2005), Iterative image reconstruction is a method of imaging reconstruction that have gained popularity in the emission tomography field, due to enhancement in image quality and other factors.

Iterative reconstruction method includes algorithms, which include sort of feedback process that allow sequential adaptations of the estimated image to the raw image/data, calculating the expected projections, then comparing the results to the raw data to maximize likelihood and estimate the so- called “correct image“. Reconstruction methods can be enhanced to include complicated and realistic system models, which result in more accurate representation of measurement of data (Hutton, B. F., Nuyts, J., and Zaidi, H,. Iterative Reconstruction Methods, 2006).

The main idea of image reconstruction in a nutshell, is to reconstruct images obtained from projection data obtained from an object from different angles, called a sinogram, which will be discussed further in the upcoming section. The images can be reconstructed from the sinogram using, for instance iterative reconstruction, finally utilizing the image for preview by doctors and/or physicians.

2.3.2. Raw Data Storage and Representation - Sinogram

The aim of image reconstruction is to recover the imaged (original) object from a set of projections, collected as a Sinogram. Sinogram is a form of raw data, which contains the projection from an imaged object as a form of angular distribution, serving as an essential part for reconstructing an image.

The sinogram is a commonly used way in displaying full projection data set of the coincidence events detected (Humanhealth.iaea.org, 2019). Thus, the raw data which is processed during the PET measurements by computer units, then sorted as sinogram, can be used to recover the original image from the projection data contained in the sinogram.

A visual flow chart representation (Zeng and Gullberg, 2010), shown in Figure 1. describing the workflow of an image reconstruction scheme using iterative algorithm, which is the main method used in this thesis.

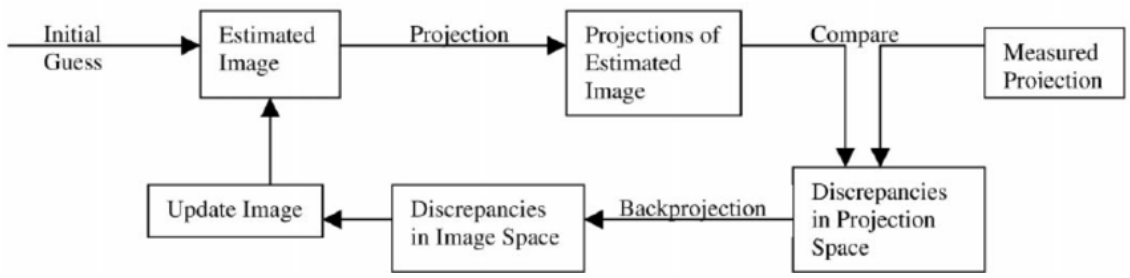


Figure 1. Flow chart of iterative image reconstruction scheme by Zeng and Gullberg, 2010.

2.4. Image Evaluation

After understanding how an image is reconstructed and formed in medical imaging, the characteristics of an image should be evaluated in terms of quality. Therefore, in this section, an introduction about image quality and criteria for assisting the image quality, common matrices used for the assessment are given. Lastly, the metrics used while conducting this for evaluation, are explained for the reader's understanding.

The methods used for assessment of image quality vary; however, they complement each other defining the essential evaluation criteria for good quality images. These methods of assessment include, calibration, resolution, uniformity, and the list go on (P. E. Kinahan and J. S. Karp 1994).

In this thesis, the methods to be focusing on mainly are, image noise and image resolution. The main reason behind focusing on these methods, is mainly to testify whether the various algorithms implemented in OMEGA v.0. produces good quality images or else, which provides readers' understanding of the resulting images and analysis.

- Image Spatial Resolution:

Image resolution in general terms, refers to the number of pixels in an image, can be defined by, for instance height and width of the image. Depending on the screen size, the resolution of the screen is set, in order to assure good quality image.

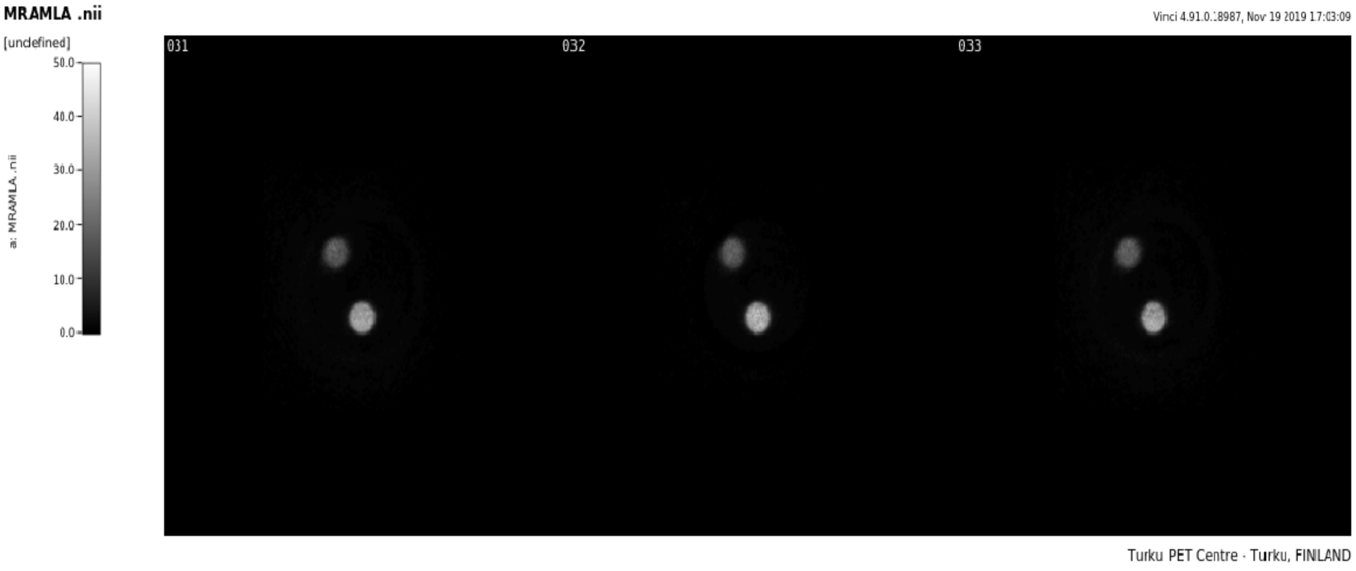
Essentially, resolution can be defined as the ability of an image or imaging system to differentiate an object of size. Thus, larger the resolution, the better is the ability to differentiate small objects from an image. The ability to differentiate high contrast small objects, as well as to distinguish close objects are characteristics of good quality, and desired image. The components affecting resolution, are like the ones mentioned earlier of image resolution in a general form; which are mainly, size of the display matrix, pixel size and zooming contrast. The size of the pixels of the display affects the resolution of the image, if it is greater than the resolution native resolution, thus effecting the image quality.

The image resolution is influenced remarkably by the platform used for reconstruction, yet sometimes zooming factor beyond a certain level cannot improve internal image resolution, resulting eventually in consequences affecting the final image quality. Spatial image resolution is refereeing to the smallest found details in an image.

Common way to measure the spatial resolution in medical image reconstruction, is by performing Full-Width-at-Half-Maximum (FWHM) worth of Point Spread Function (PSF). In this thesis, a line profile measurement is used during the analysis phase to analyze the resolution, thus the quality of the reconstructed images. In a nutshell, sharper and higher life profile should give larger FWHM, which is basically the distance between the two points where intensity is half of the maximum, when its results are small, resulting better image quality overall.

- Image Noise:

The higher the noise in an image, the lower the image quality. There are many sources of image noise in medical imaging, caused by statistical variation in the number of photons emitted and the characteristics of an imaging system. Noise in imaging techniques, can be defined, for example, by measurement of the standard deviation of a region in the image. Some of the elements affecting noise detection as well as low contrast include, detector efficiency and system sensitivity when using PET. Reconstruction algorithm is also another important factor; where certain algorithms might result in lower noise than others. An example of noise in MRAMLA algorithms in shown in (Picture 2).



Picture**2** . Noise pattern resulted from the computed algorithm MRAMLA, using Vinci v4.91 software

3. AIM AND MOTIVATION

This thesis is motivated by the enhancement of reader's knowledge, in a general form, on the topic of image reconstruction in the medical radiology field, as well as introduce the open source software OMEGA as an image reconstruction tool to be used during later stages by scientists in Turku PET Centre. Main aim of this thesis is to study and evaluate the usage of OMEGA as an open source software for PET image reconstruction. OMEGA is an open source, MATLAB software that is used for reconstructing images from PET scanner.

The software was developed by a team in Kuopio, Finland, is freely available (https://github.com/villekf/OMEGA) and is updated regularly. Understanding how the software is installed, and the status of running the data and testing of the software performance, is the aim of this study. The software OMEGAv.0 aims to reconstruct any PET sinogram or raw data. The algorithms implemented need to be tested for functionality.

Software OMEGA v.0 used while conducting this thesis, uses 15 algorithms for computing different instances for running image reconstruction. The evaluation done is from a user-class perspective; including installation ease and procedure, testing algorithms functionality and data analysis, this includes knowing the state of the version used in this thesis with its capabilities.

The evaluation criteria are focusing on eight main parts:

- Ease of installation and download.
- Software usability.
- Software reusability.
- Evaluation at a user-case perspective, from a non-scientific user.
- Testing and computing software algorithms for functionality.
- Addressing errors.
- Efficiency of software use and flexibility.
- Software accuracy.
- Software purpose fulfilment.

4. MATERIAL AND METHODS

4.1. Introduction

The material used in this thesis are obtained from various resources; physical books, e-books and peer reviewed articles from well-known journals, such as IEEE for instance, were the main resources used for research enhancement. This thesis provides overview about open source software usage in radiology and medical diagnostics, focusing on mainly the OMEGA software that is underdevelopment; used for imaging reconstruction. New interventions or innovations are not being brought with this thesis, however, evaluating of the existing software - OMEGA and testing its current algorithms were the main objectives.

Research methods applied in this research paper had an experimental approach. Completion of various testing instances while interacting with the software, analysis and validation of the resulted data were the main concentration of this thesis to enhance the existing software's quality. The implement of the research method took an agile approach; the sprint length was two weeks, and during each sprint questions or clarifications were possible to achieve the desired result by the end of each sprint.

4.2. Evaluation of Installation Procedure

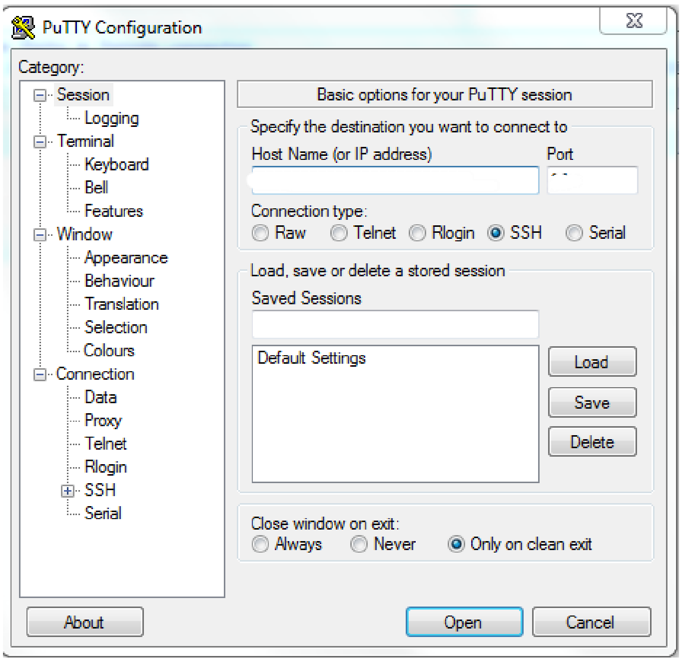
Beside OMEGA's availability as an open source software, OMEGA has great potential for support and maintenance of data. OMEGA, in this stage of experimentation is an under-development project; nevertheless, it is provided as open source and can be accessed freely as well as interacted with and performing changes to.

In this section of the thesis, step-by-step installation process of OMEGA and material used throughout each stage, will be discussed to help reader obtain a clear image of what is OMEGA and what are its functionalities. As mentioned in previous chapter, the aim of this thesis is to evaluate OMEGA from a user class, along with performing data testing and analysis. Each methodology or tool used while conducting this thesis, is indicated with its version and reference material as well.

PuTTy.exe:

The starting point is on the local computer, using Putty.exe, version 8.36 (Shown in Picture3). PuTTy.exe is the SSH and telnet client, and an OS software (https://www.putty.org/).

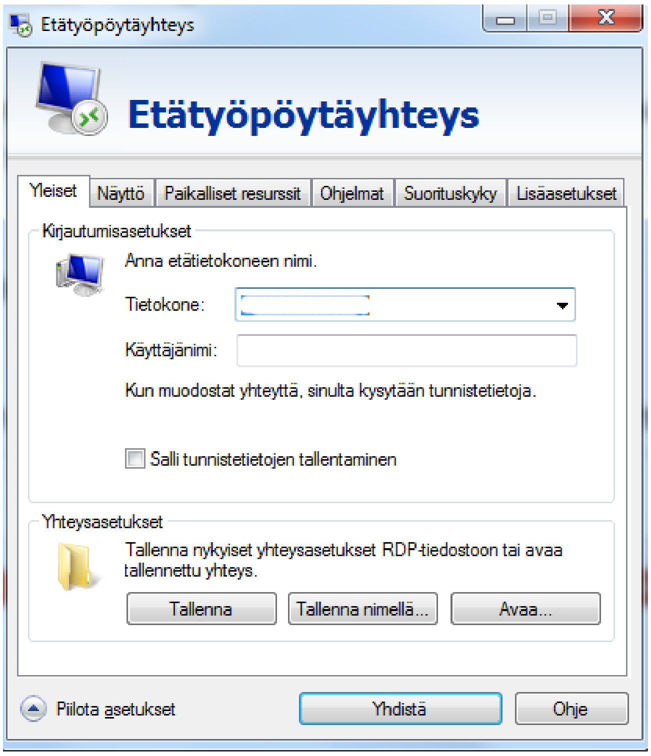
The version used while conducting this thesis is released in September.29.2019, used on Windows platform. PuTTY has a clear interface and relatively easy to use. In order to access the remote Linux server, simple steps were taken; firstly, providing the Host name, by inserting the correct IP address and login credentials, only for first item login, then finally clicking Open button for configuration. A Command-line window will pop-up, where correct username and password need to be inserted, then simply pressing on the Entre key ↲ from the keyboard to successfully finish configuration.



Picture 3. PuTTy Configuration Window

Windows-RDP:

The application Windows Remote Desktop Protocol, was needed to be able to access Linux server remotely, it is the same as running dual operating systems (OS) simultaneously, and the software version used while conducting is retrieved on September.26th.2019, Windows version 7 Enterprise, 2009. After opening the application (Shown in Picture 4), accessing the remote server needs the correct correspondent IP address. After inserting the correct IP address, click "Yhdistää" in Finnish or "Connect" in English.



Picture 4. Windows.7 RDP Application

LX-Ubuntu:

The main framework used while conducting this thesis, for procedures and applications done using the software OMEGA, is Linux Ubuntu 18.04.3 LTS. After the first couple of steps are followed successfully, Linux remote server can be automatically accessed. In this step, the second log in window on the Linux server is accessed. Inserting login credentials is needed to be able to finally access Linux remote server desktop to perform different applications on OMEGAv.0, starting with installation of the software package itself.

MATLAB:

MATLAB is a computer programming language, as well as a numerical computing environment provided by MathWorks (https://www.mathworks.com/). It is the main framework used to perform testing, and data analysis of the OMEGAv.0 data. The MATLAB version used while conducting this thesis is R2017a. To access and install OMEGA, similarly to any open-access software, the files have been cloned from a repository straight to Linux machine. The repository used for providing OMEGA is easily accessible from GitHub (https://github.com/villekf/OMEGA).

While conducting this thesis, until the final stages of experimenting, a new version of OMEGA was released, version 1.0. However, the data presented in this thesis has been constructed using the previous version of OMEGA v.0, due to the contradiction with the experiment timeline.

- **To install OMEGA to the Linux remote server:**

1. Clone files using LX-terminal or simply download the Zip-folder format from GitHub repository to the Linux server (accessed following previous steps), using the link: https://github.com/villekf/OMEGA
2. Open the OMEGA folder found in LX server.
3. Export MATLAB Path to the file downloaded, from LX-terminal Inserting the correspondent path, location and MATLAB version using: PATH=~/opt/bin:$PATH then press Enter key ↲ from keyboard.
4. A new line will appear in the LX-terminal, simply write MATLAB with no spacing nor capital letters, because LX is case sensitive, and then press the Entre key ↲ from keyboard.
5. MATLAB software automatically launches after seconds.

- **To install OMEGA software package:**

1. First important notice is to make sure that all folders and subfolders are being included to the MATLAB workspace. If not, include all folders and subfolders by simply right click on the OMEGA software folder, from the MATLAB framework, and navigate to Add to path, then Selected Folders and Subfolders to include all needed files. Navigate to *install_mex.m* from Source file, then choose Run from the Editor menu from the navigation bar. This file is necessary to compile all the needed files automatically, using GCC compiler. The GCC version OMEGA uses during conducting this experiment is 4.9. However, the GCC version MATLAB uses is 7.4. Therefore, for error prevention, navigate to all the .cpp files to be compiled and make sure that an extra library is added or add it in manually. The library meant is: #include <cstdint> or <stdint.h>, which is a part of the type support library, performing a simple fix/prevention of the GCC related error.

2. Make sure all files are compiled successfully to be able to continue to the next phase, which is reconstruction.

- **To perform image reconstruction (nongate data file used mainly throughout this thesis):**

1. Navigate to *main_nongate.m* file found in OMEGA folder.
2. Run the file by clicking the Run button from the Editor menu.
3. File *main_nongate.m* needs PET Sinogram or list-mode data *.mat file to be chosen for the function to operate; therefore, a window will pop to choose files from.
4. Click on example folder.
5. Choose any .mat file. This thesis was conducted using mainly the file, *Cylindrical_PET_example_sinogram_combined_static_200x168x703_span3.mat* and the file *Cylindrical_PET_example_measurement_cylpet_example_static_raw_ASCII.mat*
6. After choosing the *.mat file of choice, press Open to start the reconstruction process.

Following these steps, insures ease of software usage and installation. After reconstruction process is done with no errors; which the most time consuming, further evaluation procedures can be performed easily, such as algorithms testing and reconstructed image analysis.

4.3. Initial Testing Performed in MATLAB

OMEGA is a work in progress software using only test data that is included with the software at this stage of development. To test the different algorithms, a clear understanding of each is needed; furthermore, this thesis represents evaluation given by a non-scientific user, which is a user class evaluation. Performing algorithms testing is needed, in order to collect and analyses the data during later stages of this thesis.

Collecting the data by testing algorithms was performed using MATLAB, implied and performed with external code in OMEGAv.0. The OMEGA version interacted with is the version prior to 1.0. Although the newer version 1.0. can save the results in multiple format, the older version interacted with did not; therefore, there was a need for external code to be added.

For performing analysis and reading images using Vinci version 4.91 or Carimas version 2.10, NIFTI format files are needed; more about the mentioned software packages and their use will be discussed in the initial analysis sections of this thesis. In order to test each algorithm, and errors where being generated and needed fixing before passing to the net stage, the procedure took several weeks. Testing of the algorithms in MATLAB can be classified under two main method, Method1 and Method2.

Method1:

Method1 is the first approach taken for testing algorithms functionality. In order to collect data resulting from reconstructing and computing all algorithms, Method1 implies computing all algorithms to compute at the same time, automatically, by setting the option *single reconstruction* found in *main_nongate.m* file option.single_reconstruction = false; to compute all algorithms at one instance; that is by changing the *false;* value to true; then save the file using the disk Save icon found in Editor menu, and finally saving the output, to prevent overwriting the results.

Method1 could not be tested, and that is due to a missing function in the software package interacted with (OMEGA v.0). The underlying code was not modified nor changed from the original source.

Method2:

After performing the changes using Method1, and not being able to test its functionalities as mentioned in the previous section, an alternative method was considered. Method2 implies computing the algorithms in an old-fashion way, and that is by going through each of the algorithms individually changing each outcome value and start testing from there.

First step taken is to change the values from false; to true*;* for each algorithm being computed, then perform testing by noticing if file crashes, or continues computing normally while reporting results. Before starting changing algorithms values, the function option.single_reconstruction needed to be set to its original value from true; to false*;* in order to stop OMEGA from computing files automatically, to perform the changes manually for each individual algorithm. However, Method2 is more time consuming, as the algorithms needed to be tested individually. Advantages of Method2 include its ability to identify the crashing reasons, and pin-point the exact error and its origin.

Summary:

After performing both methods for testing, Method2 was found more successful than the Method1. A certain error pattern was noticed while running both methods, in one of the algorithms named MRMLA, or Modified Row-Action Maximum Likelihood Algorithm, the program crashed and even in debug mode, code cannot skip to test the next algorithm.

Testing each algorithm separately progressed well, now the issue of saving the outcome as NIFTI images arises. NIFTI images are crucial for data analysis (Nifti.nimh.nih.gov, 2019). Both methods provided further understanding of OMEGAv.0 mechanism and functionality. Testing algorithms is one of the most time-consuming steps; however, it is the key important step for building evaluation foundation. In Table1. a simple Working, Not working and Error flow for each algorithm of the 15 is presented, with focus on algorithms producing images or not. For the fourth algorithm tested, RAMLA or Row-Action Maximum Likelihood Algorithm, the algorithms were successfully computed, however it did not result any image, and more details about other algorithms is presented in the table. Images produced will be discussed in detail, in the upcoming section of this thesis.

4.4. Initial Analysis with Data Included

During this section, evaluation of the pre-analysis as well as post-analysis phases will be discussed through. The input data during the testing phase, is in *.mat format for MATLAB. Extracting images in NIFTI format was needed for visualizing the data, as well as for performing data analysis with other programs than MATLAB, as there are many software's available to support NIFTI format.

After each code modification during the testing phase, the file *visualizes_pet.m* found in OMEGA folder, was run to perform the visualized image or figure of the algorithm being computed, by reviewing the value of the data obtained in a visual result mode; however, this functionality was not yet available. An alternative way was considered, which is to extract the images in *.nii format file after each instance to a separate directory. Another MATLAB script was created, which creates the new directory where images can be saved later in *.nii format. The procedure is as following: the added code collects the data after testing results of different algorithms are obtained, loads the *.mat file, then inspects the working algorithms from the testing instances.

In this stage, the working algorithms, were eight in total (Table 1). During this phase of testing, a For loop was added in the added MATLAB file (Program1), with a purpose to extract both the image properties, such as image size, then extract the actual image and visualize it.

The For loop also included, setting datatype, origin as *default [0 0 0],* description for the image, and lastly making the actual NIFTI structure in the destination folder, setting name and automatically saving the output.

Program1 "For Loop" for extracting and saving images from the functional algorithms

```
for alg_ind=1:length(algorithm)

% Extract image properties
image_properties = pz{end,1};

% Extract image and visualize one slice
img = pz{algorithm(alg_ind)};
figure;imshow(img(:,:,5,end),[])

% Extract last iteration
img_3d=img(:,:,:,end); % 4th dimension is the iteration

%% Save the extracted image as NIFTI
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Calculate voxel size
voxel_size=[image_properties.Nx/image_properties.FOV,image_properties.
Ny/image_properties.FOV,image_properties.Nz/image_properties.axial_FOV
];

% Set origin
origin=[0 0 0]; % default

% Set datatype
datatype=512; % uint16

% Set description
description=algo_char(algorithm(alg_ind));

% Make NIFTI structure
```

```
 nii = make_nii(uint16(img_3d), voxel_size, origin, datatype);

% Set filename
filename=char(strcat(nifti_db,description,' .nii '));

% Save the NIFTI file
 save_nii(nii, filename);

end
```

Due to the testing instances and their initial result, the new code improvements were possible. After running each instance, the number of .nii image files was six in total corresponding to number of working algorithms.

Creating images to appear in a logical form was critical for moving forward to the analysis phase of this thesis. Considering the methodology approached in this research, experimenting with different approaches and techniques was needed to perform reliable results. Moving on from the testing instances to the data analysis; there are different sorts of approaches or analytical methods considered while conducting this thesis.

The resulted images from *.nii file format, were exported to the image analysis software Vinci and Carimas. The software Vinci version 4.91, along with Carimas 2.10 (Turkupetcentre.fi, 2019), for performing imaging analysis. The Vinci software stands for Volume Imaging in Neurological research and has multiple features such as image co-registration and Region of Interest (ROI) analysis. Its main objectives are visualization and analyzing of data generated from a tomographic system, such as PET. The software Vinci is expandable, dense and runs well on different systems (Nf.mpg.de, 2019).

The subtle differences between one image and another might not be sharply visible to the unaided eye, therefore quantitative analysis is needed to clarify the differences and record outcome results. Code implemented and used for analysis, can be found throughout this thesis.

In this thesis, there were mainly two sorts of data analysis done. First being VOI or ROI analysis, where the is profile analysis. Visual data analysis is performed on PET images that were reconstructed in earlier stages. During this section, VOI analysis was performed using software Vinci version 4.91 as well as using the software MATLAB version R2017a.

4.4.1. Statistical Volume of Interest Analysis (VOI) using Vinci

The first analysis experimented with VOIs, to perform reliable results from the images. The software recommended to be used at this phase is Vinci 4.91. Experimenting was key in understanding how to perform correct image analysis to produce reliable results. The data interacted with at this point is test data. The test data is from pseudo scanners measured from simulated phantoms, which are installed automatically when installing the OMEGA software.

The phantom in the included test data does not move, which made it easier to create cylinders for the VOI analysis. In OMEGA's case, VOI or ROI analysis were used a perform measurements of the pixel data inside a volume defined by a VOI; therefore, providing statistical measurements, such as mean and standard deviation to measure how the images differed from each other.

Loading images to Vinci 4.91 software:

1. Download the software Vinci 4.91 on Linux remote server, and make sure the professional use license is included.
2. From Applications menu in LX server, navigate to Graphics, then choose Vinci 4.91
3. Multiple windows will appear automatically, this is how the software opens.
4. Navigate to window <default> Load View: NIFTI Image.
5. Make sure the Plugin image type is set correctly in NIFTI Image format.
6. Select files to load, multiple files can be added by clicking the button Browse.
7. Make sure the path where images are saved is correct, in this thesis the file where NIFTI *nii images were saved is Nii_images.
8. Click Load for loading the images to the software after choosing the correct *.nii files.

After successfully adding images to Vinci 4.91 software, moving to the next step, which is image analysis. Using *VOI Define* that can be found from toolbox in the bottom of the window (horizontally) under the loaded images, with the hammer icon. From the new window VOI define, choosing New, then create new cylinder for each of the reconstructed images, to measure the VOI statistical analysis. Next step is to click on *show all...* under VOI statistics to start comparing the reference image value with the VOI image automatically and compute all statistical values. Average value, standard deviation (SD), SD percentage and co-variance are the main sections focused on to measure the image difference in this thesis. Using software Vinci

4.91 for initial statistical and VOI analysis did not successfully provide meaningful results. The software generated automatically over twenty different VOIs, which made it difficult and nearly impossible to recognize which VOI or cylinder is being measured. Which is why an alternative software was needed to perform accurate statistical analysis and result in a successful VOI data analysis.

4.4.2. Statistical Volume of Interest Analysis (VOI) using MATLAB

As discussed in the previous section, VOI analysis is needed for measuring the difference between each image and assess whether the images differed or not. An alternative approach was considered; which is to create the same analysis, however this time using code on MATLAB. MATLAB has great potential for all sorts of scientific analysis, such as, VOI analysis and visual analysis. Even though other technologies might be used for this purpose, in this thesis, MATLAB was chosen for testing, evaluation, visual and VOI analysis.

To perform VOI analysis using MATLAB, a script needed to be created to be able to run and produce accurate results for each reconstructed image. The purpose behind the script for performing VOI analysis on MATLAB, is to obtain the statistical data, such as mean, standard deviation SD and its percentage, from all the reconstructed images *.nii and save the results into a .txt file that can be later exported as an Excel file for further computations if needed.

Since the software OMEGA is developed, installed, validated and tested using MATLAB R2017a, it was only logical to try and use the same version MATLAB scripts for conducting data analysis as well. Furthermore, additional scripts were created to calculate the exact parameters needed from the images reconstructed, to conduct the VOI analysis and produce the statistical data values without generating any extra unnecessary data.

The script contains a For loop for all images (shown in Program2), where the image is defined and read for VOI calculations. As mentioned in previous section, phantom does not move in the six reconstructed images, therefore, the loop contained two cylinders which were created automatically for each slice of the image. Furthermore, list of coordinates is needed in order to view the reconstructed image and the cylinders in a logical form.

Program2 Performing Volume of Interest Analysis (VOI) using MATLAB

```
% Get all reconstructed images
image_files=dir('*.nii');

for i=1:size(image_files,1)

    % Define image to read
    img_file=image_files(i).name;%'OSEM.nii'; % change this value

    % Create directoy and file name to save data
    path_to_save=strcat('./VOI_Results/',img_file(1:end-4),'/');
   % mkdir(path_to_save)
    csv_title=img_file(1:end-4);

    % Read NIFTI file
  nifti_img=read_nifti(img_file);

    % Initialize list of coordinates for the VOI
    x=[48,81];
    y=[64,81];
    z=[21];
    start_slice_uni=1+5;
    end_slice_uni=63-5;

    % Show the phantom with centre of the VOI
    figure;imshow(nifti_img(:,:,z),[])
    hold on
    plot(x(1),y(1),'*')
    hold on
    plot(x(2),y(2),'*')
    legend('VOI1 centre','VOI2 centre')

    % Automatic cylinder VOI calculation
    cylinder_VOI_1 = make_uni_cyl([x(1),y(1)],start_slice_uni,
end_slice_uni, nifti_img, 1);
    cylinder_VOI_2 = make_uni_cyl([x(2),y(2)],start_slice_uni,
end_slice_uni, nifti_img, 1);

    % Make VOI analysis
```

```
[VOI_1_mean,VOI_1_sd,VOI_1_sdpercent,VOI_1_max,VOI_1_min]=extract_VOI_
values(nifti_img,start_slice_uni,end_slice_uni,cylinder_VOI_1);

[VOI_2_mean,VOI_2_sd,VOI_2_sdpercent,VOI_2_max,VOI_2_min]=extract_VOI_
values(nifti_img,start_slice_uni,end_slice_uni,cylinder_VOI_2);

    % Save the results as .csv
    % Write CSV file for analysis

fid=fopen(strcat(path_to_save,csv_title,'_uniformity','.txt'),'w');
    fprintf(fid,'%s,','VOI1 Mean','S.D.','Maximum','Minimum','%STD');

fprintf(fid,'\n%g,%g,%g,%g,%g,\n',VOI_1_mean,VOI_1_sd,VOI_1_max,VOI_1_
min,VOI_1_sdpercent);
   fprintf(fid,'%s,','VOI2 Mean','S.D.','Maximum','Minimum','%STD');

fprintf(fid,'\n%g,%g,%g,%g,%g,',VOI_2_mean,VOI_2_sd,VOI_2_max,VOI_2_mi
n,VOI_2_sdpercent);
    fclose(fid);

end
```

Next step is to show the phantom correspondent to center of the VOI analysis, this step is followed by automatically calculating the VOI cylinders. In this phase of analysis, the cylinders are being generated. After the previous steps are done, the generated files from the VOI analysis need to be saved in a form of .csv or .txt for further analysis if needed. In a nutshell, this procedure can be considered handier while using MATLAB R2017a, rather than using Vinci 4.91 for VOI analysis. MATLAB provides further control and accuracy when interacting with the data, increasing the repeatability and reproducibility of further analysis performed on the images.

4.4.3. Profile Analysis Using MATLAB

After experimenting with the previous analysis methods, which include statistical analysis in two different software, a different analysis focusing on the visual aspect is considered. Analysis done visually in medical imaging provide higher understanding to the situation. The main idea behind creating visual analysis is to create a comprehensive overview of the actual images resulted and evaluate their quality.

The visual analysis used in this thesis is Profile analysis, or plot analysis to measure the differences between each image accurately without viewing unnecessary data. Profile analysis, or plot analysis shows the differences in test results, in a graphical presentation – visualizing the changes in pixel values as both shapes and numbers, which helps in identifying whether two or more of test results, in this case images are being represented and visualized. The profile analysis serves as a very rough estimate of FWHM, as explained in Theoretical background, section 2.4 under: image Spatial Resoution section.

To perform profile analysis on the resulted images, extra code (Program3) is needed to be implemented. The platform used mainly for writing the code as well as performing the profile analysis is MATLAB R2017a. The code started with defining the *.nii image to be read, adding list of coordinates for the profiles, showing the mean of all the profiles tested, then finally calculating and saving profiles.

Program3 Performing Profile Analysis using MATLAB

```
% Define image to read
img_file='OSEM .nii'; % changeable value to any *.nii image

% Create directoy to save data
path_to_save=strcat('./Results/',img_file(1:end-4),'/');

% Read NIFTI file
nifti_img=read_nifti(img_file);

% Add list of coordinates for the profile for each patient
x=[30,98];
y=[50,91];
z=[21];

% Show the phantom and start / end of profile
figure;imshow(nifti_img(:,:,z),[])
hold on
plot(x(1),y(1),'*')
hold on
```

```
plot(x(2),y(2),'*')
legend('Start','End')

% Add constant profile offsets?
offset_x=[-1,0,1];
offset_y=[0,0,0];

% Calculate and save profiles
[profile]=calculate_line_profile_linux(nifti_img,x,y,path_to_save);

% Shown the mean of the all profiles
myfig=figure;
plot(mean(profile(:,:,:),3),'.-b')
lgd=legend(img_file(1:end-4));
lgd.FontSize = 18;
myfig.Children(2).FontSize=14;
xlim([0 size(profile,1)])
ylim([0 40])
ylabel('Activity (a.u)','FontSize',18)
xlabel('Profile location (pix)','FontSize',18)
title('Mean profile over all slices')
```

Profiles were created on MATLAB after obtaining non-accurate results from Vinci software. Since the phantom in the included data does not move, the results were shown in a way where the starting point and the ending point of the measuring profile is created for each phantom and computed for each reconstructed image from the algorithms.

In Program2, a line from the code used to save the profiles after calculations is previewed; *Profile_linux* was used because the OS interacted with at this stage is Linux and for the resulted profiles to be viewed on Linux OS. X and Y are the list of coordinates used for each profile during analysis and shown in resulting figure as well.

The main operation to ensure the accuracy of data at this stage was is to show the mean of all profiles resulted after each calculation in a final figure. The procedure was repeated for all images reconstructed. By performing this last step, the implementation section of this thesis is finalized. In the upcoming chapters, results will be discussed furthermore, providing more understanding regarding evaluation assessment criteria.

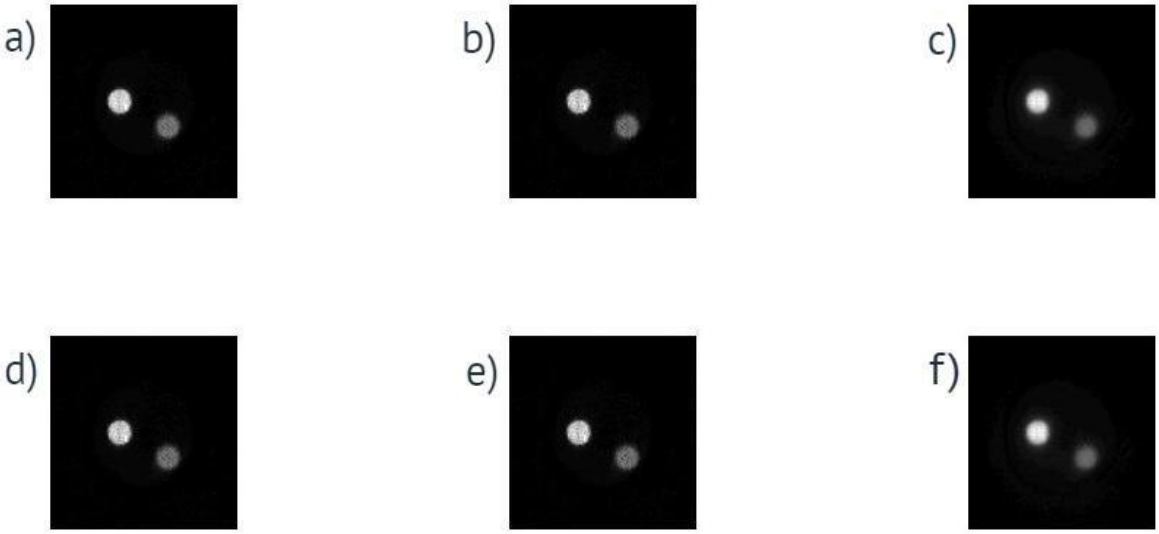
5. RESULTS

This section of the thesis is mainly focusing on addressing the results obtained after performing initial testing analysis of the included data using OMEGAv.0, by experimenting with the different software, for data analysis and testing, such as Vinci 4.91 and MATLAB R2017a. This chapter is divided into two main sections, starting with the images resulting after the testing phase of OMEGAv.0 using MATLAB R2017a, and the second section includes the results obtained from image analysis, VOI and Profile analysis, using software MATLAB R2017a.

5.1. Images of the Evaluation data

The testing instances were the main time consumers while conducting this thesis, however they were the concrete ground for building the rest of the operations. After performing several testing instances for the fifteen algorithms computed for images reconstructing, and even though eight of the fifteen algorithms were successfully run, only six algorithms resulted in images; these are the images being shown, compared and analyzed through this thesis (Table1).

Considering all performed testing methods, steps and new modifications to the code that were added, optimal results delivered were successfully reached. All reconstructed images (shown in picture 5) showed difference corresponding to the algorithms tested, noticing that six algorithms of the eight functional algorithms reconstructed enough images.



Picture **5**. The reconstructed images resulted from six successfully computed algorithms. Images in order: a)OSEM, b)ECOSEM, c)COSEM, d)ACOSEM, e)MRP-OSL, f)MRAMLA.

Table 1. Results from Testing of the Algorithms

Algortithm name	**Working**	**Not working**	Errors
MLEM		-	Undefined function or variable ‘pz_osem_apu’ Error in reconstruction_main line 1108 and main_nongate line 332
OSEM	-		
MRMLA	-		
RAMLA	With no resulting images		
ECOSEM	-		
COSEM	-		
ACOSEM	-		
MRP-OSL	-		
MRP-BSREM	With no resulting images		
Q.P. OSL		-	Undefined function or variable ‘padding’. Error in reconstruction_main line 865 and in main_nongate line 332
Q.P. BSREM		-	Undefined function or variable ‘padding’. Error in reconstruction_main line 865 and in main_nongate line 332
L-OSL		-	Undefined function or variable ‘padding’. Error in reconstruction_main line 865 and in main_nongate line 332
L-BRSEM		-	Undefined function or variable ‘padding’. Error in reconstruction_main line 865 and in main_nongate line 332
FMH_OSL		-	Undefined function or variable ‘pz_pad’. Error in reconstruction_main line 935 and in main_nongate line 332
W.M OSL		-	Undefined function or variable ‘pz_pad’. Error in reconstruction_main line 942 and in main_nongate line 332
In total = 15 Algorithm	8 2 = No images	7	

The reconstructed images at this phase might look the same, however they are not. The images differed slightly, for example, in the noise or texture of each image or the overall quality from one figure to another. In the upcoming section of this thesis, furthermore details about analysis results are presented, for providing the evidence and the accurate calculations on how the images differed from one another will be discussed.

5.2. Evaluation of the Image Quality

After performing validation of the included data and successfully obtaining results from computing the algorithms, the procedure continued to obtain further sorts of evaluation. In order to provide evidence of accuracy and evaluate each image's quality, different approaches were considered. In this section, results of the profile analysis performed in MATLAB R2017a.

- Profile Analysis Results:

During this stage, the resulting analysis were accurate and provide further clarity on whether the images reconstructed are similar or not. Since the procedure was repeated for all .nii images, differences were highly noticeable. The image ACOSEM.nii resulting from the algorithm ACOSEM, which stands for Acceleration Conversion OSEM, had different results than OSEM.nii where the profiles were slightly changed, however the mean profile over all phantom slices took a different, rather higher peak on the scale of activity.

The image resulted from the image MRP-OSL.nii, resulted from algorithm MRP-OSL, which stands for Median Root Period with One Step Late. The analysis results of this image were rather interesting, whereas all images the phantom does not move, however the profiling measurements differed slightly.

MRAMLA algorithm stands for Row-Action Maximum Likelihood Algorithm reconstruction, resulting profiles of similar quality of MRP-OSL. Finally, the images resulted from computing profile analysis for the algorithms ECOSEM and COSEM. The second algorithm COSSEM stands for complete data OSEM. ECOSEM on the other hand is responsible for convergent COSEM, which is the algorithm discussed earlier. The algorithm initials stand for Enhanced Convergent OSEM.

All analysis of the algorithms describes above are represented in Figure2. As noticeable, almost all the main algorithms that were successfully built and run during initial testing phase are related to one another.

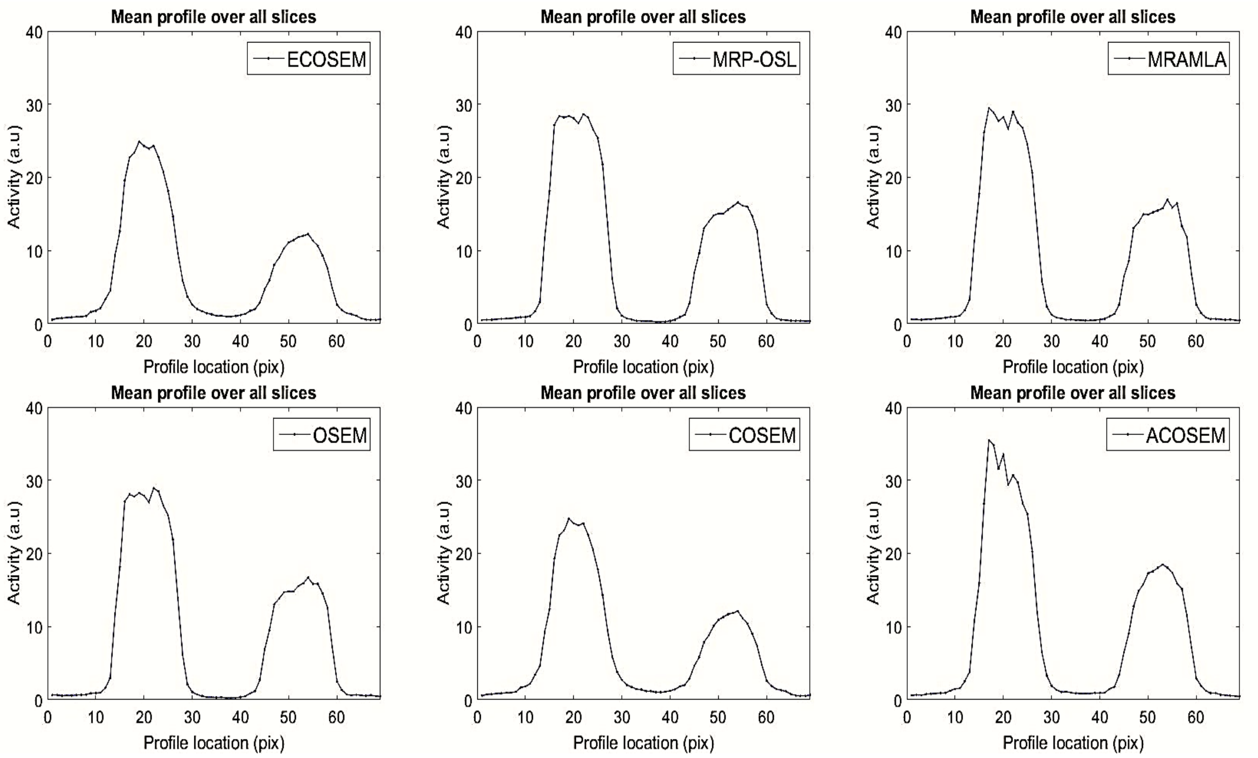


Figure 2. Visual representation of the Profile analysis from the six reconstructed images.

In conclusion, Profile analysis provides comparisons between the algorithms. The data in this thesis was plotted in a way that showed the activity points on the y-axis and the profile location (pix) on the x-axis.

- ROI / VOI Analysis

The results obtained from MATLAB R2017a were more logical than the data set represented by Vinci 4.91, and fulfilled their experimental research objectives. To provide better understanding, the following table (Table2), shows all the VOI results from the six reconstructed images; showing mean, standard deviation, SD percentage, as well as maximum and mean value of each image in a statistical format.

Table 2. VOI Analysis

ACOSEM					
VOI1	Mean	S.D.	Maximum	Minimum	%STD
	26.1094	10	53	3	40.1018
VOI2	Mean	S.D.	Maximum	Minimum	%STD
	13.5178	5.5	32	2	40.4356
COSEM					
VOI1	Mean	S.D.	Maximum	Minimum	%STD
	19.4704	7.3	36	3	37.7095
VOI2	Mean	S.D.	Maximum	Minimum	%STD
	8.87434	3	19	2	34.2135
ECOSEM					
VOI1	Mean	S.D.	Maximum	Minimum	%STD
	19.7033	7.4	36	3	37.5695
VOI2	Mean	S.D.	Maximum	Minimum	%STD
	9.05989	3.1	19	2	34.3022
MRAMLA					
VOI1	Mean	S.D.	Maximum	Minimum	%STD
	24.5927	9.5	48	2	38.7683
VOI2	Mean	S.D.	Maximum	Minimum	%STD
	12.8051	4.9	29	1	38.1773
MRP-OSL					
VOI1	Mean	S.D.	Maximum	Minimum	%STD
	24.8851	9.7	50	1	38.8258
VOI2	Mean	S.D.	Maximum	Minimum	%STD
	13.1123	5	31	1	38.3187
OSEM					
VOI1	Mean	S.D.	Maximum	Minimum	%STD
	24.9311	9.4	46	2	37.7496
VOI2	Mean	S.D.	Maximum	Minimum	%STD
	13.1888	4.8	27	1	36.5516

6. DISCUSSION

While experimenting with different analysis software, accurate results have been reached and will be discussed in detail to provide evaluation and further understanding of the mechanism accuracy of OMEGAv.0 as an image reconstruction software. During this section, thesis limitations, possible further implementations, and results will be discussed, to bring the experimental research conducted throughout this thesis together, discussion regarding the key findings conducted, and their meaning in practice.

The outcome of the reconstructed images, resulting from computing all the working algorithms when performing VOI analysis, which are eight in total, showed that only six algorithms computed to fulfill their purpose, and that is to provide an image. Such results can be considered as key for taking OMEGAv.0 to the next development step; where instead of having six algorithms constructing images, all the fifteen algorithms computing simultaneously to provide reconstructed images.

When discussing OMEGAv.0 as code, clearly it has high-quality documentation material, providing detailed explanation to all code aspects; however, there is a need for a more simple user guide of the documentation of OMEGAv.0 code, which can be contradicting the main purpose of documentation.

Results found while performing analysis, on the other hand, were key for understanding and evaluating the evaluation of the images and the data in general; nevertheless, if the working algorithms that were computed did not provide any images, reaching analysis phase would have not been possible.

- **Limitations of the thesis**

Implementing and conducting this thesis started with testing algorithms; followed up by data collecting in different forms, such as, in the form of statistical or visual data, and eventually evaluating the data resulted.

In this section of discussion chapter, limitations while conducting the research on evaluating OMEGAv.0 software are discussed through. While conducting this thesis and throughout each sections of research, whether external material or achieved internal results, limitations were experienced.

The limitations of this thesis include inspecting limitations and processing strategies, as well as motivation and other possible limitations. Ability to perform such daunting tasks in short period of time needs motivation. Will-power and motivation concepts are key and affecting strongly on this thesis process. It is fruitful to constantly interact with, and learn challenging technologies and their usages for academic progression, although there was a steep learning curve.

The time frame and other factors as well, imposed limitations on the amount of data analysis and data evaluation implemented throughout this thesis. Therefore, boundaries were important for remaining on schedule and meeting set deadlines.

Running the software smoothly from the beginning was not the case while conducting this thesis and experimenting with OMEGAv.0, many errors occurred which can be considered normal to begin with. However, while running the testing instances, some of the algorithms took almost a full work-day or two, therefore, time was one of the limitations as well. Luckily, limitations regarding ethics or data usage were not experienced throughout conducting this thesis.

One of the limitations occurred at late stages of conducting this thesis, and after obtaining results, is the release of the new OMEGA version 1.0 officially. This thesis, on the other hand, is discussing the general evaluation of OMEGA as a software, using the version .0 not version 1.0.

Finally, and due to the limitation of time, initial testing of the new version of OMEGA v.1.0 was done; therefore, the initial testing on OMEGA v.1.0. is not presented in this thesis, to not contradict the results found using OMEGA version.0.

- **Future implementations and developments**

As discussed earlier, possibility of enhancing the OMEGA software is available, and one of the ways to achieve this, is by simply enhancing, for instance, documentation. OMEGAv.0 software and code documentation was extensively and explicitly provided by previous developers in a detailed format, besides the test instances and the data obtained while conducting this thesis.

The documentation provides detailing information on every individual file, algorithm and their functionality. Since the information is critically important, high-quality, useful and applicable, it can be daunting and difficult to grasp.

The possible future implications on documentation for instance, is to provide the same information of interacting with the software and its functionalities, for reference; however, enhance or create software manual that can be easily understood by users. This way it is easier for users to find reference on some of the issues faced during interacting with OMEGA v0., however, not get overwhelmed with the amount of information there is at least on the manual.

Another future development feature to be experienced with, is evaluating with Turku PET Centre's own data using OMEGAv.0 instead of using included data. Reconstruction using "real" data from an experiment, for example, can enhance the software, raises its possibility to reach medical imaging and image reconstruction software markets quicker; since its accuracy is tested with both types of data, included and/or added to OMEGA.

Information about the reference activity in the phantom is sure to provide gold standard reference in comparison of the algorithms. Last and not least, the final enhancement to the software is a basic (or advanced) user interface, to enhance the OMEGA software experience. At this stage, OMEGA is accessed throughout the software MATLAB, providing only back-end review and resulting images in a form of pop-up windows as figures or pictures, which can be only opened in MATLAB, or as .txt files, that can be opened in any text-file program or exported as an Excel file.

Furthermore, creating a simple UI, where for example each user can create a profile for adding, removing, and editing data would be a great feature to add, for enhancing OMEGA's experiment. The UI can be in a form of independent program application, or web-based application. Even though this process might seem difficult; however, after reaching optimal results in computing all algorithms and having OMEGA running with no errors, constructing images and fulfilling its purpose, creating a user experience or a UI will seem less daunting and easily applicable.

7. CONCLUSION

The main purpose of this thesis was to study and evaluate OMEGAv.0. usability, sustainability, testing its algorithms and analyzing the reconstructed images resulted, to ensure software purpose fulfillment. By performing algorithm testing of software OMEGAv.0, results were successfully obtained. The algorithms resulted in six images that were mainly used for analysis, which were discussed throughout this thesis. Also, visual and statistical analysis were crucial for evaluating OMEGAv.0 as an image reconstruction software.

The objectives of evaluation OMEGAv.0. such as: ease of installation, usability, software purpose fulfilment, and other objectives were successfully met and fulfilled during this thesis; for example, installation using the step-by-step format can be easily followed and reused, thus fulfilling the goal of evaluating ease of use, of the OMEGAv.0 software.

The evaluation carried throughout this thesis, is from a none scientific, user-case perspective. The evaluation was carried out by testing and analysis, over a variety of software, focusing on mainly Vinci 4.91 and MATLAB R2017a. The data dealt with while conducting this thesis, is mainly included data.

OMEGAv.0 as a software has promising potentials. Being able to test OMEGAv.0.'s features provided further understanding of its general mechanism, thus providing the ability of software evaluation. The initial data test and evaluation provided results, that were discussed in previous sections; furthermore, the significance of the results provide further understanding and possible continuation of evaluation phase to validation and further testing in the future of OMEGAv.0. software's life cycle.

Since this thesis is conducted on a scientific software, by a non-scientific user, the findings and procedures can be easily followed by non-scientifc users as well as by scientific users. The data resulted while conducted this thesis can be reproduced, and repeated, providing good software sustainability, which is key when developing OS software. In conclusion, OMEGAv.0. as an open source software, is capable of reconstructing images to a high extent and can be easily used by scientists within Turku PET Center.

8. REFERENCES

1) Dowsett, D., Kenny, P., Johnston, R. (2006). The Physics of Diagnostic Imaging. London: CRC Press, Retrieved September 15, 2019, at https://doi.org/10.1201/b13462
2) Hutton B.F., Nuyts J., Zaidi H. (2006) Iterative Reconstruction Methods. In: Zaidi H. Quantitative Analysis in Nuclear Medicine Imaging. Springer, US, Retrieved September 14, 2019, at https://link.springer.com/book/10.1007/b107410
3) Basic Nuclear Medicine Physics. (2013). Essentials of Nuclear Medicine Physics and Instrumentation, 1–20. Retrieved September 15, 2019, at http://doi.org/10.1002/9781118473481
4) Adcox, B. K. and M. (2015). PET scan: Definition, Purpose, Procedure, and Results. Retrieved September 16, 2019, at https://www-pub.iaea.org/MTCD/Publications/PDF/Pub1617web-1294055.pdf
5) Turkupetcentre.net. (2019). *TPC - Image reconstruction*. Retrieved September 16, 2019, at http://www.turkupetcentre.net/petanalysis/image_reconstruction.html
6) Humanhealth.iaea.org. (2019). *Human Health Campus - Basic PET physics and instrumentation (Part 1)*. Retrieved September 16,2019, at https://humanhealth.iaea.org/HHW/NuclearMedicine/Webinars/Basic_Nuclear_Medicine/Basic_PET_physics_and_instrumentation_(Part_1)/index.html
7) Kinahan, P. E., & Karp, J. S. (1994). Figures of merit for comparing reconstruction algorithms with a volume-imaging PET scanner. *Physics in Medicine and Biology*, *39*(3), 631–642. Retrieved September 16,2019, at http://doi.org/10.1088/0031-9155/39/3/024
8) Villekf. (2019, May 7). villekf/OMEGA. Retrieved September 17, 2019, at https://github.com/villekf/OMEGA
9) Turku PET Centre Finnish National Research Institute for the use of short-lived positron emitting isotopes in the field of medical research. Retrieved September 23, 2019, at http://turkupetcentre.fi/
10) D. Margan and S. Čandrlić, "The success of open source software: A review," *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Opatija, 2015. Retrieved 25 September, 2019, at http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7160503&isnumber=7160221

11) gnu.org. GNU-Licenses. Retrieved September 25, 2019, at https://www.gnu.org/licenses/licenses.en.html
12) “MATLAB.” MathWorks, Retrieved September 25, 2019, at https://www.mathworks.com/products/matlab.html.
13) Turkupetcentre.fi. (2019). *Carimas*. Retrieved September 27, 2019, at http://turkupetcentre.fi/carimas/download/
14) Nifti.nimh.nih.gov. (2019). *NIfTI: — Neuroimaging Informatics Technology Initiative*. Retrieved September 27,2019, at https://nifti.nimh.nih.gov/
15) Tohme, M. S., & Qi, J. (2009, June 21). Iterative image reconstruction for positron emission tomography based on a detector response function estimated from point source measurements. Retrieved September 27, 2019, at https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2901798/#R30
16) Mahmic-Kaknjo, M. (2017). Retrieved 27, September 2019, at http://cejpaediatrics.com/index.php/pt/article/view/282/pdf
17) The Central European Journal of Paediatrics, 13(1), 42–45. Retrieved October 10, 2019, at http://doi.org/10.5457/p2005-114.168
18) Winkler, A. M., et al. “The NIFTI File Format.” Brainder., 6 Sept. 2017, Retrieved October 10, 2019, at https://brainder.org/2012/09/23/the-nifti-file-format/.
19) Mcauliffe, Matthew & Lalonde, Francois & McGarry, D.P. & Gandler, William & Csaky, Karl & Trus, Benes. (2001). Medical Image Processing, Analysis & Visualization in Clinical Research. Retrieved October 20, 2019, at https://www.researchgate.net/publication/3908878_Medical_Image_Processing_Analysis_Visualization_in_Clinical_Research
20) Download PuTTY - SSH and telnet client for Windows. Retrieved November 5, 2019, at https://www.putty.org/
21) Humanhealth.iaea.org. (2019). *Human Health Campus - 3D image reconstruction*. Retrieved November 5, 2019, at: https://humanhealth.iaea.org/HHW/MedicalPhysics/NuclearMedicine/ImageAnalysis/3Dimagereconstruction/index.html
22) Medical image reconstruction: A conceptual tutorial - Scientific Figure on ResearchGate. Retrieved 7 Nov, 2019, at https://www.researchgate.net/figure/Flow-chart-of-iterative-image-reconstruction-scheme_fig13_285944235

23) Vinci Software. Retrieved November 7, 2019, at http://www.nf.mpg.de/vinci3/doc/vinci-about.html#sec-3-