

YaIT, Yet another ImageTool 0.5

Anne Seppänen
Calle Laakkonen
PET-keskus, Turku

29.7.2003

TURKU PET CENTRE INFORMATION TECHNOLOGY REPORT SERIES

REPORT TPCIT0006



1 Käyttöohje

1. **Ohjelman käynnistys:** YaIT ohjelman voi käynnistää CDE:n taustavalikosta Analysing Programs → YaIT.0.5.beta tai komentoriviltä komennolla YaIT.0.5. Version 0.3 voi käynnistää taustasta samaan tapaan ja komentoriviltä komennolla YaIT.
2. **Konfiguraatiotiedosto:** YaIT lukee muutamia arvoja konfiguraatiotiedostosta. Jos käyttäjä haluaa ottaa käyttöön omat oletusarvot, voi konfiguraatiotiedoston /usr/local/config/yait.cfg kopioida omaan kotihakemistoon nimellä .yait.cfg ja muokata tiedoston itselleen sopivaksi.
3. **Kuvatiedoston avaus:** Kuvatiedosto avataan valitsemalla File → Open. Kuvatiedoston valinnan jälkeen annetaan dialogiin ne matriisit (frame plane), jotka halutaan näytölle, esimerkiksi: "1-6 15,17-20". Jos näytölle halutaan esimerkiksi:
 - kaikki matriisit, voi dialogin jättää tyhjäksi ja valita vain ok.
 - kaikki tasot tietyn freimiltä esim 4, dialogiin voi kirjoittaa pelkän luvun 4 ja valita ok.
 - staattisesta kuvasta tietyt tasot esim. 4-10, dialogiin on kirjoitettava ensin frame eli "1 4-10" ja valitaan ok. Huomaa välilyönti lukujen 1 ja 4 välissä.
 - dynaamisesta kuvasta freimit 3-5, joilta kaikilta tasot 5,6,7,9,12,13 ja 14, kirjoitetaan dialogiin "3-5 5-7,9,12-14". Huomaa jälleen ainoa välilyönti lukujen 5 välillä.
4. **Zoomaus:** Hiiren vasemmalla näppäimellä valitaan ensin kuvamatriisi, joka halutaan zoomata. Zoomattava matriisi saa punaiset kehykset. Valitun matriisin frame ja plane näkyvät myös ikkunan yläosassa olevissa tekstikentissä. Zoomattavan kuvan voi valita myös ikkunan yläreunassa olevista nuolista. Valitaan Options → zoom → * zoomauksen mukaan. Ikkunan yläreunassa sijaitsevista (frame ja plane) nuolista voidaan myös siirtyä helposti seuraaviin matriiseihin. Options → show selected näyttää aiemmin valitut kuvamatriisit uudelleen. Options → Select again avulla näytettävät kuvamatriisit voidaan valita uudelleen.
5. **ROI:n piirto:** ROI (Region Of Interest) tarkoittaa kuvassa olevaa mielenkiintoaluetta. Kun kuva on zoomattu ja halutaan piirtää ROI, valitaan ROI → Draw → *.

- Trace ROI piirretään hiiren vasemman näppäimen avulla ja lopetaan oikealla näppäimellä, joka samalla sulkee ROI:n.
- Suorakaide ROI piirretään klikkaamalla ensin alkupiste ja sen jälkeen loppupiste hiiren vasemmalla näppäimellä. Hiiren oikea näppäin peruuttaa piirron.
- Ympyrä ROI:n keskipiste merkitään hiiren vasemmalla näppäimellä, jonka jälkeen hiirtä liikutetaan ja painetaan jälleen hiiren vasenta ympyrän kehällä. Hiiren oikea näppäin peruuttaa piirron.
- Ellipsi ROI piirretään kuten ympyrä ROI.

ROI:n piirtämisen jälkeen ohjelma kysyy käyttäjältä ROI:n nimeä. Samaa nimeä ei voi antaa saman matriisin kahdelle eri ROI:lle. Saman kuvapaketin eri matriiseilla voi olla samannimisiä ROI:ta. Jos ROI:lle ei anneta nimeä, se hylätään.

6. **ROI:n aktivointi hiirellä:** Painetaan hiiren vasenta näppäintä ROI:n päällä. Aktivoitu ROI piirtyy punaisella ja ikkunan alalaitaan tulee tietoa kyseisestä ROI:sta.
7. **ROI:n siirto hiirellä:** Valitaan ROI → Draw → Move, jonka jälkeen voi valita ROI:n sitä klikkaamalla. Valittu ROI vaihtaa väriä ja sitä voi nyt liikuttaa hiirellä. ROI siirtyy uuteen paikkaan painamalla hiiren vasenta näppäintä. Oikea näppäin peruuttaa siirron. Valitun ROI:n pystyy myös siirtämään painamalla 'm' näppäintä.
8. **ROI:n pyöritys hiirellä:** Valitaan ROI → Draw → Rotate, jonka jälkeen voi valita ROI:n sitä klikkaamalla. ROI:ta voi nyt pyörittää liikuttamalla hiirtä. Hiiren vasen näppäin vahvistaa pyörityksen ja oikea näppäin peruuttaa. Huomaa että vain Trace ROI:ta pystyy pyörittämään! Jos haluat pyörittää muita ROI:ta, täytyy ne ensin muuttaa Trace ROI tyyppisiksi.
9. **ROI:n kopiointi:** Yhden tai usemman ROI:n voi kopioida sellaisenaan tai peilikuvana valitsemalla ROI → Draw → Copy tai ROI → Draw → Mirror copy. Listasta valitaan kopioitavat ROI:t ja tekstikenttään kirjoitetaan matriisit (frame plane), joihin ROI kopioidaan, esim. "1-3 2,5-8,10". Samaan matriisiin ei voi piirtää tai kopioida kahta samannimistä ROI:ta. Peilikuvana kopioitavan ROI:n nimeen liitetään päätte, esim. _mirror. ROI:n nimen voi myöhemmin käydä muuttamassa valitsemalla ROI → List.
10. **ROI:n monistus:** Voit monistaa valitun ROI:n valitsemalla ROI → Draw → Double. Seuraava hiiren klikkaus valitsee ROI:n. Valitusta ROI:sta ote-

taan kopio ja voit nyt siirtää sitä hiirellä. Vasen hiiren nappi asettaa ROI:n uudelle kohdalleen, oikea palauttaa alkuperäisen ROI:n päälle.

11. **ROI:n muutos Trace ROI tyyppiseksi:** Suorakaide, ympyrä tai ellipsi ROI:n pystyy muuttamaan Trace ROI tyyppiseksi. Tämä on tarpeen jos haluat esim. pyörittää ROI:ta. Valitse ROI jonka haluat muuttaa ja valitse valikosta ROI → Draw → Convert to trace. YaIT varmistaa vielä haluatko muuttaa ROI:n tyyppiä. Muutosta ei voi peruuttaa.
12. **ROI:n rajaamien pixelien näyttäminen:** Valitse ROI ja paina Control näppäintä. ROI:n rajaamat pixelit vaihtavat väriään. Väri palautetaan normaaliksi painamalla controllia uudestaan.
13. **ROI:n poisto:** Valitun ROI:n voi poistaa painamalla 'Del' näppäintä.
14. **ROI-tiedosto:**
 - ROI-tiedoston voi avata valitsemalla ROI → File → Load. Tiedoston valinnan jälkeen ohjelma kysyy, mitkä kaikki ROI:t halutaan hakea kyseisestä tiedosta ja mille kuvamatriiseille valitut ROI:t haetaan. Dialogista valitaan ROI:t. Mikäli haettavat ROI:t halutaan eri kuvamatriisille kuin mihin ne alunperin on piirretty, kirjoitetaan dialogin alaosaan frame(t) ja plane(t).
 - ROI:t voi tallettaa tiedostoon valitsemalla ROI → File → Save. Ensin valitaan tiedoston, jonka jälkeen ohjelma antaa uuden dialogin, josta käyttäjä valitsee ne ROI:t, jotka haluaa tallettaa. Mikäli ROI-tiedosto on jo olemassa, ohjelma kysyy käyttäjältä lisätäänkö talletattavat ROI:t tiedoston perään vai hävitetäänkö edellinen ROI-tiedosto.
 - ROI → Save in valikosta määrätään kumman tutkimuksen frame ja plane numeroiden mukaan ROI:t tallennetaan. Tällä on vaikutusta silloin kun on ladattu kaksi tutkimusta joiden vierekkäisten kuvien frame ja/tai plane numerot eivät täsmää.
 - Valitsemalla ROI → List ohjelma näyttää kaikki piirretyt ja haetut ROI:t, joiden nimiä voi muokata tai joita poistaa.
 - Valitsemalla ROI → Show in matrix view, ROI:t saa näkyviin jo matriisi listassa.
15. **Värit:** Värikarttaa voi vaihtaa valitsemalla Options → color → map → *.Värien skaalausta voi vaihtaa valitsemalla joko Options → color → scale → automatic tai Options → color → scale → fixed. Värikarttaa voi

skaalata klikkaamalla yläpalkissa näkyvää palettia. Värikartan minimiarvo määrätään klikkaamalla paletin yläreunaa ja maksimiarvo klikkaamalla alareunaa. Mikäli näytölle valituissa matriiseissa on negatiivisia pikselin arvoja, ohjelma ilmoittaa käyttäjälle tästä. Valitsemalla Options → color → scale → No negative values negatiiviset arvot nollataan.

16. **Kahden tutkimuksen lataaminen:** Ennen kuin toinen tutkimus voidaan ladata, ensimmäinen tutkimus pitää olla ladattuna. Kun ensimmäinen tutkimus on ladattu, tulee File valikossa esiin Open secondary study. Sitä klikkaamalla voidaan toinen tutkimus avata. Options → Select secondary again valikosta voidaan valita kakkos tutkimuksen matriisit uudelleen. Jos tutkimuksissa vierekkäin olevissa kuvissa on eri frame/plane numerot, voidaan ROI → Save in valikosta valita kumman tutkimuksen mukaan ROI:t tallennetaan.

17. **Muuta:**

- Menuihin liittyvä pikavalinta on Ctrl-x, missä x on menussa alleviivattu kirjain. Esimerkiksi File → Open toimii myös näppäinvalinnalla Ctrl-o.
- Kun ensimmäinen kuvapaketti on analysoitu, ja halutaan hakea toinen kuvatiedosto, on suositeltavaa valita ensin File → Clear.
- Näytöllä olevan kuvan voi tulostaa paperille valitsemalla File → Print.

18. **Ongelmia:**

- ROI:n piirto jatkuu (joskaan ei kovin toimivana) vaikka menisi zoom tilasta pois tai vaihtaisi kuvaa. (Eli jos aloittaa ROI:n piirron, se pitäisi myös lopettaa)
- Zoomattujen kuvien tulostus landscapena ei toimi oikein: kuva ei ole keskitetty.
- YaIT saattaa kaatua jos kuvat eivät ole neliön muotoisia.
- Molempien tutkimuksien kuvien pitää olla samankokoisia

2 Dokumentointi

YaIT 0.5 ohjelman käyttöliittymä on tehty Motif toolkit 2.1:n avulla. Ohjelma käyttää seuraavia kirjastoja: libpet, libm, libXm, libXt, libX11. Scripti `/usr/local/bin/YaIT.0.5`, joka on Bourne-shellissä, tarkistaa löytyykö käyttäjän kotihakemistosta omaa konfiguraatitiedostoa `.yait.cfg`. Scripti kutsuu varsinaista ohjelmaa `/usr/local/bin/YaIT.0.5.beta` ja antaa kutsussa oikean konfiguraatitiedoston. Tulostukset ohjataan käyttäjän kotihakemistoon lokitiedostoon nimeltä `.yait.log`.

Yait 0.4 koostuu tällä hetkellä 30 c-kielisestä lähdekoodi- ja otsikkotiedostosta parista sekä kahdesta omasta otsikkotiedostosta.

Otsikkotiedostot

1. colortables.h

- Värikarttojen määrittely. Muuttujat static const tyyppisiä. Tämän tiedoston tietojen perusteella generoidaan oikeat värikartat.

2. yait.h

- structuurien ym. määrittelyä.

Lähdekooditiedostot externinä määriteltyjen funktioiden prototyytit löytyy `.c` tiedostoa vastaavasta `.h` tiedostosta static funktioita ei käytetä kyseisen `.c` tiedoston ulkopuolella.

1. close_file.c

- **extern void clear_display_area(void)** Poistaa kuvat ja tummentaa valikot (Load secondary, print ym...)

2. color.c

- **extern void color_cb(Widget widget, XtPointer client_data, XtPointer call_data)** funktio päivittää käyttäjän valitseman värikartan sekä varsinaiseen `display_area`an että `color_bariin`. Kutsutaan kuvan päivittäviä funktioita zoomin arvon mukaan.
- **extern void color2_cb(Widget widget, Xtpointer client_data, XtPointer call_data)** vastaava kuin `color2_cb`, mutta päivittää kakkos tutkimuksen värikartan.

- **extern void color_scale_cb(Widget widget, XtPointer client_data, XtPointer call_data)** funktio asettaa panels.scale muuttujan arvon 1 tai 2 riippuen, onko käyttäjä valinnut automaattisen vai fiksatun skaalauksen sekä kutsuu kuvan päivittäviä funktioita. Automaattisessa skaalauksessa jokainen kuvamatriisi skaalataan erikseen. Fiksatussa skaalauksessa etsitään ensin kaikkien näytettävien matriisien minimi ja maksimi arvot, jonka jälkeen kaikki matriisit skaalataan tälle välille.
- **extern void set_colorscale_cb(Widget widget, XtPointer client_data, XtPointer call_data)** skaalaa värikarttaa kun käyttäjä klikkaa palletti widgettiä.

3. cursor.c

- **extern void my_SetCursor(Widget w)** funktioissa vaihdetaan kursori erilaisiin muotoihin.
- **extern void my_ResetCursor(Widget w)** funktiossa palautetaan kursori tavanomaiseksi nuolikursoriksi.

4. defaults.c

- **extern void set_defaults()** funktiossa asetetaan oletus- ja alkuarvoja globaaleille muuttujille mm. värikarttoihin liittyville muuttujille.
- **extern void reset_panel()** funktiossa asetetaan uudestaan joitakin oletusarvoja. Tätä funktiota kutsutaan mm. kun käyttäjä valitsee **File** → **Clear**.

5. deviation.c

- **extern void calc_dev(IMG *imgpack, int matnum, ROI *roi, float *mean, float *dev)** funktio laskee ROI:n mukaiselta alueelta kuvamatriisin pikseleiden keskiarvon ja -hajonnan.

6. double_roi.c

- **extern void double_roi_cb(Widget w, XtPointer client_data, XtPointer call_data)** funktiossa otetaan kopio valitusta ROI:sta ja

aloitetaan ROIn siirto.

7. export.c

- **extern void export_cb(Widget w, XtPointer client_data, XtPointer call_data)** export valikon callback. Kutsuu export_dialogia oikealla tiedostoformaatilla.
- **extern void export_image(Widget widget, XtPointer client_data, XtPointer call_data)** exporttaa kaikki ladatut kuvat tai vain zoomatun kuva(pari)n. Exporttaus dialogi kutsuu tätä kun painetaan OK:ta.

8. export_dialog.c

- **extern void export_dialog(ExportType type)** Avaa exporttaus dialogin sopivalla tiedostotyypillä.
- **static void export_pushed(Widget w, XtPointer client_data, XtPointer call_data)** Export dialogin OK callback. Kutsuu export_imagea.

9. file_cb.c

- **extern void destroy_widget(Widget w, XtPointer client_data, XtPointer call_data)** funktio tuhoaa widgetin ja kutsuu cursorin resetoivaa funktiota.
- **extern void mselection_cb(Widget w, XtPointer client_data, XtPointer call_data)** funktiossa luodaan dialogi, jossa käyttäjä valitsee ne kuvapakan matriisit, jotka haluaa näytölle.
- **extern void file_cb(Widget widget, XtPointer client_data, XtPointer call_data)** funktiota kutsutaan, kun käyttäjä valitsee jonkin File menun toiminnon. File → Open: luodaan tiedostojen selailuun tarkoitettu dialogi (filebrowser), jossa käyttäjä voi valita haluamansa kuvatiedoston. File → Print: kutsutaan funktiota **view_print**. File → Clear ja File → Quit: tarkistetaan onko ROI_listassa tallettamattomia ROI:ta ja ilmoitetaan niistä käyttäjälle. Mikäli tallettamattomia ROI:ta ei löydy, palautetaan alkuarvoja tai suljetaan ohjelma.

- **extern void clear_again(Widget dialog, XtPointer client_data, XtPointer call_data)** funktioon päädytään, jos käyttäjä haluaa tyhjästä muistia tallettamattomista ROIsta huolimatta.
- **extern void quit_again(Widget dialog, XtPointer client_data, XtPointer call_data)** funktioon päädytään, jos käyttäjä haluaa sulkea ohjelman tallettamattomista ROIsta huolimatta.

10. help_cb.c

- **extern void helpmenu_cb(Widget widget, XtPointer client_data, XtPointer call_data)** Help valikon callback.
- **void help_cb(Widget widget, XtPointer client_data, XtPointer call_data)** funktiossa luodaan dialogi, jossa näytetään (char*)client_data.

11. images.c

- **extern void refresh_image_list(int studyN)** päivittää studyN:ään kuuluvat kuvat muuttamalla 8 bittinen kuva näytön kuvasyvyyteen screen_format_ximage_update funktiolla. Jos studyN on pienempi kuin 0, kaikki tutkimukset päivitetään.
- **extern void update_image_list(Study *study)** tarkistaa onko käyttäjän haluamia frameja ja planeja kuvassa, poistaa vanhat kuvat (jos on) ja kutsuu add_images funktiota.
- **extern void clear_images(void)** poistaa kaikki kuvat ja niiden widgetit
- **extern void add_images(ScaleType scale, char replace)** tekee kuvat raasta datasta ja luo niihin kuuluvat widgetit. Jos replace on != 0, vanhoja widgettejä ei poisteta vaan niille vain asetetaan uusi data.
- **extern ImageList *get_image_list(void)** palauttaa osoittimen linkitettyyn listaan joka sisältää ladatut kuvat.
- **extern YaIT_Image *get_selected_image(void)** palauttaa osoittimen valittuun kuvaan.
- **extern ImageList *find_image_pos(YaIT_Image *image)** Palauttaa osoittimen kuvalistan alkioon joka sisältää annetun kuvan.

- **extern YaIT_Image *find_image(int study,int frame,int plane)** Palauttaa kuvan joka täsmää annetut hakuehdot. Jos study<0, kumpi tahansa kelpaa.
- **extern int select_image_n(int study,int frame,int plane)** Asettaa kuvan joka täsmää hakuehdot valituksi. Käyttää find_image funktiota kuvan löytämiseen.
- **extern int find_frame_plane_index(Study *study,int frame,int plane,int *iframe,int *iplane)** Etsii missä kohtaa kyseinen kuva sijaitsee datassa. Asettaa iframe ja iplane muuttujiin oikeat indexit.
- **extern void show_color_table(int d_limit, int u_limit,int cmapN)** päivittää paletti widgetit sekä kuvat.
- **extern void update_matrix_view(int studyN)** piirtää uudelleen kaikki kuvat.
- **extern void get_minmax(Study *study,float *min,float *max)** Hakee suurimman ja pienimmän arvon tutkimuksesta.
- **extern void message_area_info(void)** näyttää tietoa viestipalkissa valitusta kuvasta.
- **extern void message_area_info_study(Study *study)** näyttää tietoa viestipalkissa tietyistä tutkimuksesta. (Muuten kuin normaali message_area_info, mutta ei näytä valittua plane/frame numeroa)
- **extern YaIT_Image *get_2nd_zoom_image(YaIT_Image *image)** Palauttaa kuvan joka näkyy annetun kuvan vieressä (zoomauksessa) kun 2 tutkimusta on ladattu.

12. interpolation_cb.c

- **extern void interpolation_cb(Widget w, XtPointer client_data, XtPointer call_data)** funktio päivittää interpoloinnin päälle/pois käyttäjän valitessa Options → Interpolation → (None/Linear).

13. list_cb.c

- **extern void list_cb(Widget widget, XtPointer client_data, XtPointer call_data)** funktiota kutsutaan, kun käyttäjä on valinnut ROI → List, joka näyttää listan kaikista ROI_listan ROIsta ja joita voi nimetä uudelleen tai joita voi poistaa listalta.

- **extern void read_roi_cb(Widget widget, XtPointer client_data, XtPointer call_data)** funktiossa luetaan ja tulkitaan käyttäjän antamat matriisit, joille haettavat ROI:t kirjataan.
- **extern void copy_roi_cb(Widget widget, XtPointer client_data, XtPointer call_data)** funktiossa luetaan ja tulkitaan käyttäjän antamat matriisit, ja kopioidaan valitut ROI:t niille.
- **extern void mirrorcopy_roi_cb(Widget widget, XtPointer client_data, XtPointer call_data)** funktiossa luetaan ja tulkitaan käyttäjän antamat matriisit, ja kopioidaan valitut ROI:t peilikuvana niille.
- **extern void selectall_cb(Widget widget, XtPointer client_data, XtPointer call_data)** valitsee kaikki ROI:t.

14. main.c

- määrittellään globaalit muuttujat.
- **int main(int argc, char *argv[])** funktiossa luodaan ohjelman toplevel-niminen application shell, jonka päälle luodaan muut widgetit.
- **static void fill_control_panel(Widget control_panel)** funktiossa luodaan "spinboxit" sekä väripalkki pääikkunan yläreunaan.
- **static void create_menus(Widget menubar)** funktiossa luodaan kaikki menut.

15. message_dialog.c

- **extern void message_dialog(char *message, char *title)** funktiossa luodaan dialogi, jossa näytetään kutsussa saatu viesti.
- **extern void info_dialog(char *message, char *title, int index)** funktiossa luodaan dialogi, jossa näytetään kutsussa saatu viesti. Dialogin ok-nappulaan on liiteyty callback-funktioksi joko `clear_again` tai `quit_cb` funktio, riippuen mistä kusu alunperin tuli.
- **extern void error_dialog(char *message)** funktiossa luodaan dialogi, jossa näytetään kutsussa saatu virheilmoitus.
- **extern void fatal_error_dialog(char *message)** funktiossa luodaan dialogi, jossa näytetään kutsussa saatu vakava virheilmoitus.

Käyttäjälle annetaan mahdollisuus sulkea ohjelmaa tai yrittää jatkaa.

- **extern void quit_dialog(char *message)** funktiossa luodaan dialogi, joka kysyy, haluaako käyttäjä todellakin sulkea ohjelman.
- **static void quit_cb (Widget dialog, XtPointer client_data, XtPointer call_data)** funktio sulkee ohjelman.

16. move_roi.c

- **void move_roi_cb(Widget w, XtPointer client_data, XtPointer call_data)** funktiossa luetaan kursorin paikka ja aloitetaan ROI:n siirto. Seuraava kutsu tähän funktioon joko vahvistaa siirron tai peruuttaa sen.

17. options_cb.c

- **void options_cb(Widget w, XtPointer client_data, XtPointer call_data)** funktiota kutsutaan, kun käyttäjä valitsee Options → Select again tai Options → Show selected. Edellisessä luodaan dialogi, johon käyttäjä voi antaa haluamansa matriisit ja jälkimmäisessä tapauksessa kutsutaan funktiota **showimage**, joka näyttää aiemmin valitut kuvamatriisit.

18. options_cb.c PetImage widgetti. tätä käytetään kuin muitakin widgettejä. PetImage tukee näitä uusia propertyjä:

- **petimageNframe** kuvan frame numero.
- **petimageNplane** kuvan plane numero.
- **petimageNstudy** kuvan tutkimuksen numero.
- **petimageNselected** onko kuva valittu tai ei (Ei liity widget traversaliin)
- **petimageNselectedCb** valinta callback. (Kun widgettiä klikataan vasemmalla hiiren nappulalla)
- **petimageNroiSelectedCb** ROI:n valinta callback. call_data sisältää valitun ROI:n.
- **petimageNmouseCb** callback jota kutsutaan kun painetaan jotain muuta hiiren nappia.

- **petimageNframeColor** valitun kuvan kehyksen väri.
- **petimageNROI** osoitin ROI_list struktuuriin. PetImage näyttää ne ROI:t joiden frame ja plane numerot täsmäävät. (sekä mahdollisesti roi_userdatan extra_matnum)
- **petimageNzoom** zoomaus kerroin.
- **petimageNroiMode** ROI:n piirtomoodi.
- **petimageNselectedRoi** osoitin valittuun ROIhin.
- **petimageNrequestRoi** kun tämän asettaa, petimage kutsuu roiSelectedCb callbackia tämän arvolla.
- **petimageNroiColor** ROI:n väri
- **petimageNselRoiColor** valitun ROI:n väri

19. print.c

- **extern void print_to_file(char *filename, PaperOrientation orientation, int paperw, int paperh, double top, double bottom, double left, double right, Boolean header)** luo PostScript tiedoston ja exporttaa matriisilistan tai zoomatun kuvan sinne.

20. print_dialog.c

- **extern void print_dialog(void)** Näyttää tulostus dialogin

21. profile.c

- **extern void profile_cb(Widget w, XtPointer client_data, XtPointer call_data)** Callback jota kutsutaan zoomatuista kuvista niinkuin ROI:n piirto callbackia. Piirtää yhden viivan ja avaa viiva-profiili ikkunan.

22. read_arg.c

- **extern void re_read_arg(Widget w, XtPointer client_data, XtPointer call_data)** lopettaa zoomauksen, poistaa vanhat kuvat ja kutsuu read_argia.

- **extern void read_arg(Widget w, XtPointer client_data, XtPointer call_data)** funktio lukee ja tarkistaa käyttäjän selection dialogissa pyytämät matriisit.
- **extern void read_matrix_sel(char *matrix_str, INT_list *frame_list, INT_list *plane_list)** funktio muuttaa käyttäjän antaman stringin kahdeksi INT_listaksi. INT_list-tyyppinen tietue, joka on määritelty libpet kirjaston intex.h tiedostossa, sisältää kaksi kenttää: *i sisältää kokonaislukuja ja nr kertoo montako kokonaisluku *i sisältää.

23. read_config.c

- **extern void read_config(int ac, char *av[])** funktio lukee ja tulkitsee konfiguraatiotiedoston.

24. roi_cb.c

- **extern void roi_menu_cb(Widget menu_item, XtPointer client_data, XtPointer call_data)** ROI valikon callback, näyttää ROI listan jos käyttäjä valitsi ROI → List tai näyttää ROI:t suoraan matrix viewissä (ROI → Show in matrix view)
- **extern void roi_draw_cb(Widget menu_item, XtPointer client_data, XtPointer call_data)** funktiota kutsutaan, kun käyttäjä valitsee ROI → Draw → *. Funktio luo uuden ROI:n ja kutsuu piirtofunktioita.
- **extern void roi_tools_cb(Widget menu_item, XtPointer client_data, XtPointer call_data)** funktiota kutsutaan, kun käyttäjä valitsee ROI → Tools → *. Funktio kutsuu muita ROI:n käsitteilyfunktioita.
- **extern void roi_file_cb(Widget w, XtPointer client_data, XtPointer call_data)** funktiota kutsutaan, kun käyttäjä valitsee ROI → File → * ja kutsuu edelleen muita funktioita.
- **extern void roi_save_cb(Widget w, XtPointer client_data, XtPointer call_data)** päivittää Save ROI to muuttujan valikon mukaan.

25. roi_dialog.c

- **extern void name_roi_dialog(char *message)** funktiossa luodaan dialogi, jossa käyttäjä voi nimetä piirtämänsä ROI:n.
- **static void name_roi_cb(Widget widget, XtPointer client_data, XtPointer call_data)** funktiossa luetaan ROI:n nimi ja tarkistetaan, onko käyttäjän antama nimi mahdollinen. Jos on, niin lisätään uusi ROI ROI-listaan, muuten kutsutaan uudelleen funktiota **name_roi_dialog**. Jos nimeä ei annettu, ROI poistetaan.
- **extern void rename_dialog(char *message, int *ind)** funktiossa luodaan dialogi, jossa käyttäjä voi nimetä ROI-listasta valitsemansa ROI:n uudelleen.
- **static void rename_cb(Widget widget, XtPointer client_data, XtPointer call_data)** funktiossa luetaan käyttäjän antama uusi nimi ROI:lle ja tarkistetaan onko nimi mahdollinen ja päivitetään.
- **extern void roi_save_dialog()** funktiossa luodaan tiedostojen selailuun tarkoitettu dialogi (filebrowser), jossa voi valita mihin roitiedoston valittavat ROI:t talletetaan. Dialogin ok-nappulan callbackiksi on asetettu funktio **select_save_roi**, jossa käyttäjä voi valita ne ROI:t, jotka haluaa tallettaa.
- **extern void check_roifile(Widget widget, XtPointer client_data, XtPointer call_data)** funktio tarkistaa, onko talletettava roitiedosto jo olemassa. Jos tiedosto on olemassa, luodaan dialogi, jossa käyttäjältä kysytään kirjoitetaanko olemassa olevan tiedoston päälle vai lisätäänkö ROI(t) tiedostoon. Jos tiedostoa ei ennalta ole olemassa, talletetaan ROI:t suoraan tiedostoon.
- **static void save_roilist_cb(Widget widget, XtPointer client_data, XtPointer call_data)** funktiossa tuhoetaan vanha roitiedosto ja kutsutaan **append_roilist_cb**:tä.
- **static void append_roilist_cb(Widget widget, XtPointer client_data, XtPointer call_data)** funktiossa lisätään ROI:t roitiedostoon.
- **extern void roi_read_dialog()** funktiossa luodaan tiedostojen selailuun tarkoitettu dialogi (filebrowser), jossa voi valita haettavan roitiedoston.
- **static void read_roifile_cb(Widget widget, XtPointer client_data, XtPointer call_data)** funktiossa luodaan dialogi, jossa näytetään kaikki roitiedostossa olevat ROI:t. Käyttäjä voi näin valita, mitkä

ROI:t halutaan hakea tiedostosta.

26. roi_functions.c

- **extern void traceroi_cb(Widget w, XtPointer client_data, XtPointer call_data)** funktiossa kerätään käyttäjän piirtämän trace ROI:n kulmapisteet.
- **static void finish_trace_roi(ROI *roi, RoiPointList *points, int count, int offset)** funktio kirjaa trace ROI:n arvot ROI-tietueeseen.
- **extern void rectangleroi_cb(Widget w, XtPointer client_data, XtPointer call_data)** funktio piirtää ja kirjaa suorakaiteen muotoisen ROI:n.
- **extern void circleroi_cb(Widget w, XtPointer client_data, XtPointer call_data)** funktio piirtää ja kirjaa ympyräroin.
- **extern void ellipseroi_cb(Widget w, XtPointer client_data, XtPointer call_data)** funktiota kutsutaan, kun käyttäjä haluaa piirtää ellipsi ROI:n.
- **static void finalize_roi(Widget w, ROI *roi)** viimeistelee ROI tietueen.

27. roilist.c

- **extern ROI *put_roi_to_list(ROI *roi, ROI_list *roi_list)** funktio kopioi ROI tietueen tiedot ROI-listaan. Palauttaa osoittimen listassa olevaan kopioon.
- **extern int get_roi_from_list(ROI *roi, ROI_list *roi_list, int i)** funktio lukee ROI-listasta ROI tietueeseen arvot.
- **extern void repair_rois(ROI_list *roi_list)** funktio korjaa ROI-den pariliitokset kahden tutkimuksen tilassa.

28. roiutils.c

- **extern char hit_roi(ROI *roi, int x, int y)** tarkistaa onko piste (x,y) roi:n sisällä.
- **extern int roi_area(ROI *roi)** palauttaa ROI:n pinta-alan.

- **extern XImage *lego_roi(Display *d, Visual *v, ROI *roi, int width, int height, int zoom)** Palauttaa XYBitmap tyyppisen XImagegen joka näyttää roin peittämät pixelit.

29. roi_rotate.c

- **extern void _rotate_roi_cb(Widget w, XtPointer client_data, XtPointer call_data)** funktiossa luetaan kursorin paikka ja aloitetaan ROI:n pyörytys. Seuraava kutsu tähän funktioon joko vahvistaa pyörytyksen tai peruuttaa sen.

30. scroll_list.c

- **extern void scroll_list_rois(void)** funktiota kutsutaan, kun käyttäjä on valinnut ROI → List. Funktiiossa luodaan dialogi, jossa listataan kaikki ROI_listan ROI:t, joita voi poistaa listalta tai nimetä uudelleen.
- **extern void select_save_roi(Widget widget, XtPointer client_data, XtPointer call_data)** funktiossa luodaan dialogi, jossa valitaan talletettavat ROI:t.
- **extern void select_copy_roi()** funktiossa luodaan dialogi, jossa valitaan kopioitavat ROI:t ja jossa annetaan matriisit, joille valitut ROI:t kopioidaan.
- **extern void select_mirrorcopy_roi()** funktiossa luodaan dialogi, jossa valitaan peilikuvana kopioitavat ROI:t. Dialogissa voi antaa loppupäätteen, joka lisätään alkuperäisen nimen perään, kun ROI muutetaan peilikuvaksi.

31. spin_cb.c

- **extern void spin_cb(Widget parent, XtPointer client_data, XtPointer call_data)** frame ja plane spinboxien callback. Vaihtaa valittua kuvaa.

32. util.c

- **extern Widget CreateActionArea (Widget parent, ActionAreaItem *actions, int num_actions)** luo dialogille 'action arean'

- **extern void unmanage_cb(Widget widget, XtPointer client_data, XtPointer call_data)** piilottaa kutsuvan widgetin. Jos client_data on asetettu, poistetaan se XtDestroyWidget kutsulla.
- **extern int find_table_entry(XmStringTable table,int count,char *entry)** etsii merkkijonotaulukosta annetun merkkijonon.

33. zoom_cb.c

- **extern void zoom_cb(Widget w, XtPointer client_data, XtPointer call_data)** funktio asettaa zoom_matrix.zoomin arvon ja kutsuu zoomauksen aloittavia funktioita.
- **extern zoom_current_image(void)** on apufunktio joka zoomaa valitun kuvan.

34. zoom_images.c

- **extern void init_zoom_images()** alustaa zoomauksessa käytetyt muuttujat ja luo widgetit. Tätä kutsutaan ohjelman käynnistyessä.
- **extern void set_zoom_image(int frame,int plane,int frame2,int plane2,int zoom,int interpolate)** piilottaa matriisilistan, laskee zoomatun kuvan ja näyttää sen.
- **extern void scale_image(int zoom,int interpolate)** skaalaa kuvan ja asettaa sen zoom widgettiin.
- **extern void zoom_stop(int refresh)** lopettaa zoomauksen ja näyttää matriisilistan. Jos refresh!=0, matriisilistan kuvat päivitetään add_images() kutsulla.
- **extern void refresh_zoom(int studyN)** päivittää zoomatun kuvan 8 bittisestä ximagesta. Jos studyN<0, kaikki tutkimukset päivitetään.
- **extern void set_ROI_drawmode(ROI *roi,int mode)** asettaa widgetit ROI:n piirtomoodiin.
- **extern PetimageWidget get_zoom_image_widget(void)** palauttaa ensimmäisen tutkimuksen widgetin.
- **extern PetimageWidget get_zoom_image_widget(void)** palauttaa ZoomImage struktuurin.

Tulevia muutoksia ja parannuksia

- Help ja about (enemmän tietoa).
- ROIItten talletuksessa roitiedoston valinta ja talletettavien ROIItten valinta voisi olla toisessa järjestyksessä.
- 8 bittisen näyttömoodin tuki