

# Variables applied – SPM/AA/qsub/...

**Tibor Auer**

MRC Cognition and Brain Sciences Unit, Methods group

# Variables revisited

---

- **Definition: a piece of information in the memory (RAM)**
  - Fast access and manipulation
  - Finite lifetime
- Scope/Visibility/Access:
  - Global (special use and explicit definition to avoid confusion)
  - Local

# Variables revisited

---

- Type (how the data is stored in the RAM):
  - Simple
    - Char: one character = 1 byte
    - Logical:
      - True = 1
      - False = 0
    - Numeric
  - Complex
    - Array
    - Structure

# Variables revisited

---

- **Numeric type**
  - Integer: (u)int8-int32
    - Number of bits
      - Memory space: 8 bit = 1 byte
      - Range: e.g.:
        - int8:  $0-\pm 2^{8-1} \text{ (sign)}-1 = \pm 127$
        - uint8:  $0-2^8 \text{ (no sign)}-1 = 0-255$
  - Floating point: double (8 byte), single (4 byte)

# Arrays applied

---

- **Sequence of ordered elements with the same (matrix) or different (cell) type**
  - An fMRI dataset: 3 mm iso, 192 mm FOV, 32 slices, 200 volumes in doubles (default):
    - 8 byte: one voxel x → 8 byte
    - x (192/3 = 64): one line x → 512 byte
    - x 64 line: one slice x → 32768 byte = 32 kB
    - x 32 slices x → 1024 kB = 1 MB
    - x 200 volumes → **200 MB**

# Structures applied

## SPM

---

- **Organised/“structured” storage of variables with any type**
  - SPM default settings (for fMRI):
    - *spm('Defaults','FMRI');*
    - *global defaults*
    - *defaults*
    - *defaults.realign.estimate* (settings for motion correction estimation)
      - *quality: 0.9000*
      - *interp: 2*
      - *wrap: [0 0 0]*
      - *sep: 4*
      - *fwhm: 5*
      - *rtm: 1*

# Structures applied

## SPM

---

- SPM job: matlabbatch
  - *matlabbatch{1}.spm.spatial.realignunwarp.eoptions*
    - *quality: 0.9000*
    - *sep: 4*
    - *fwhm: 5*
    - *rtm: 0*
    - *einterp: 2*
    - *ewrap: [0 0 0]*
    - *weight: "*

# Structures applied

## SPM

---

- SPM design: SPM =
  - xBF: [1x1 struct]: (canonical) hemodynamic base function
  - ...
  - *SPM.xBF*:
    - *UNITS:= 'secs'*
    - *Volterra: 1*
    - *Name: 'hrf'*
    - *length: 32.0625*
    - *order: 1*
    - *T: 32*
    - *T0: 17*
    - *dt: 0.0625*
    - *bf: [513x1 double]*
  - *SPM.xBF.UNITS = 'secs'*



# Structures applied

## SPM

---

- SPM design: SPM =
  - *xY*: Data
  - *Sess*: Design settings (events)
  - *xX*: Design Matrix
  - *xCon*: Contrasts (only after specifying them)

# Structures applied

aa

---

- aa settings = aap
  - Everything about your analysis
  - Built:
    - 1. aap\_parameters\_defaults\_CBSU.xml: generic settings (site-specific)
- *aap.directory\_conventions*
  - *rawdatadir: '/mridata/cbu:/mridata/csl:/mridata/camcan'*
  - *seriesoutputformat: 'Series\_%03d\*'*
  - *protocol\_fieldmap: 'FieldMapping'*
  - *protocol\_structural: 'MPRAGE'*
  - *fsldir: '/imaging/local/software/fsl/latest/x86\_64/fsl'*
  - *feesurferdir: '/imaging/local/software/freesurfer/latest/x86\_64/'*



# Structures applied

aa

---

- aa settings = aap
  - Everything about your analysis
  - Built:
    - 1. aap\_parameters\_defaults\_CBSU.xml: generic settings (site-specific)
    - 2. aap\_tasklist\_fmri.xml: task settings
- *aap.tasksettings.aamod\_realignunwarp.eoptions<sup>1</sup>*
  - *quality: 1*
  - *COMMENT: {1x7 cell} (description of the options)*
  - *sep: 2*
  - *fwhm: 5*
  - *rtm: 0*
  - *einterp: 7*
  - *ewrap: [0 1 0]*
  - *weight: ""*

# Structures applied

aa

---

- aa settings = aap
  - Everything about your analysis (*aa\_doprocessing(aap)*)
  - Built:
    - 1. *aap\_parameters\_defaults\_CBSU.xml*: generic settings (site-specific)
    - 2. *aap\_tasklist\_fmri.xml*: task settings
    - 3. User Master Script: customise settings, add data, model
  - *aap.acq\_details*: Data
    - *root*: *'/imaging/ta02/Workshop'*
    - *numdummies*: 5
    - *subjects*: *[1x2 struct]*
    - *sessions*: *[1x1 struct]*
    - ...
  - *aap.tasksettings.aamod\_firstlevel\_model.model*: Model for all subjects (1<sup>st</sup> is empty)
  - *aap.tasksettings.aamod\_firstlevel\_contrasts.contrasts*: Contrasts for all subjects (1<sup>st</sup> is empty)

# Structures applied

## qsub

---

- qsub: Queuing system results in Job folders
  - aa: they are in your home/aaworker folder
  - aa engine monitors it for you!
  
- Job1\_script.sh - qsub engine: last line shows you the qsub settings
- matlab\_metadata.mat - MATLAB Distributed Computing engine
- Task1.common.mat - Text!: Date and time of access to the job
- **Task1.diary.txt** - **Output to the Command window**
- **Task1.in.mat** - **The actual job**
- Task1.jobout.mat
- **Task1.log** - **Output to the Terminal**
- **Task1.out.mat** - **Results/Error of the job**
- Task1.state.mat - Text!: job status

# Structures applied

## qsub

---

- Task1.out.mat - The actual job
  - *createtime: 'Fri Dec 06 17:59:55 GMT 2013'*
  - *taskfunction: @aa\_doprocessing\_onetask*
  - *argsin: {[1x1 struct] 'doit' [2] [1]}*
  - *name: 'Task1'* - Name (qstat)
  - *nargout: 0*
- Actual command: *aa\_doprocessing\_onetask(argsin{1},'doit',2,[1])*
  - *argsin{1}: aap*
  - Run the second module on the first subject
    - *argsin{1}.tasklist.main.module(2).name = aamod\_get\_dicom\_structural*

# Structures applied

## qsub

---

- Task1.in.mat
  - *erroridentifier: ""*
  - *errormessage: ""*
  - *worker: [1x1 struct]*
  - *argsout: {}*
  - *errorstruct: []*
  - *finishtime: 'Fri Dec 06 18:00:09 GMT 2013'*
- worker:
  - *Host: 'node-d08.mrc-cbu.cam.ac.uk'* - The actual machine
  - *ComputerType: 'GLNXA64'*
  - *ProcessId: 31540* - ID (qstat)

# Structures applied

## Errors

---

- Error structure
  - Unhandled: *lasterror*
  - Handled: try-catch block (*test\_error\_call*)
- What was the error...
  - *identifier: 'MATLAB:UndefinedFunction'*
  - *message: 'Undefined function or variable 'this\_is\_not\_an\_existing\_function'.'*
  - *cause: {0x1 cell}*
- **We have called an un-existing/undefined/un-found function...**



# Structures applied

## Errors

---

- Error structure (*test\_error\_call*)
  - ... and where did it occurred (from bottom to top)
    - *stack: [2x1 struct]*
      - *E.stack(1)*
        - *file: 'D:\Workshop\2\_MATLAB\test\_error.m'*
        - *name: 'test\_error'*
        - *line: 3*
      - *E.stack(2)*
        - *file: 'D:\Workshop\2\_MATLAB\test\_error\_call.m'*
        - *name: 'test\_error\_call'*
        - *line: 1*- ... in line 3 of **test\_error** called by line 1 of **test\_error\_call**

# Storing variables

## Simple types and arrays

---

- **Text file**
  - Cross-platform
  - Easy to read/write
  - May require conversion to MATLAB (*importdata*, *dlmread*, *textscan*)
- **.mat file**
  - MATLAB only
  - Any type
  - Compressed
  - Native in MATLAB (no conversion required)

# Storing variables

---

- **XLM file: aa**
  - Cross-platform
  - Multi-level
  - “Inheritance”
  - Complicated to read and write actual data
  - *xmlread, xmlwrite, xml\_read* (in aa, no way back)
- **Custom: IniFile**
  - MATLAB only
  - Two-levels only
  - No inheritance (but completion)
  - Easy to read and write actual data
  - *IniFile*

# Thank you for your attention!

---

- **Questions?** – If you dare!



**I pity the fool  
who doesn't use aa**