# How to calculate the variance components

Supplementary materials to:

***Quantifying sources of variability in gait analysis***

by Kohleth Chia and Morgan Sangeux, **DOI: 10.1016/j.gaitpost.2017.04.040**

Contacts: kohleth@gmail.com, morgan.sangeux@gmail.com

## Introduction

This document describes the process to extract gait kinematics data from c3d files and to perform the statistical analysis described in *Quantifying sources of variability in gait analysis*, K. Chia and M. Sangeux, Gait & Posture 2017, on new data.

## Extracting data from c3d files

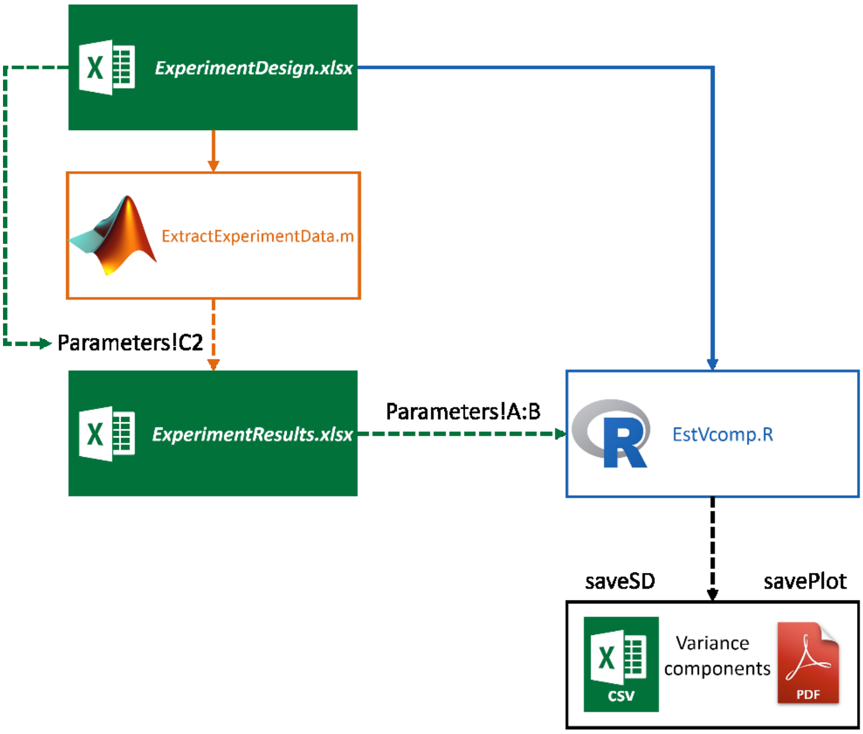
The first step consists in extracting the kinematic curves. We assume the motion capture files have been saved in c3dfiles. Although it would be possible to adapt the script to extract data from other files than c3d files, this would require significant changes and has not been tested.

Data extraction is performed in Matlab and we provide the script *ExtractExperimentData.m*. The script requires two inputs: *XLfilename* and *isdebugon*. *XLfilename* is the full path of an Excel file that specifies the experiment design and location of c3d files [default: ExperimentDesign.xlsx]. *XLfilename* provides all the information required to extract the data, as well as all the information regarding the design of the experiment, as per the example provided: subject, assessor, session and trial. *isdebugon* value is 1 [default] or 0, if 1 the script will only issue warnings when it encounter an issue so that the user may scan for all issues initially. However, it is advised to set this to 0 once all issues have been resolved. In this case any remaining issue will raise an error and the script will stop.

The btk (biomechanical toolkit, [1]) library for matlab is required and can be downloaded here. The btk and several other Matlab functions are required so make sure you include their respective folders in the Matlab path.

If the script run without errors, it creates a XL file [*ExperimentResults.xlsx*] with one spreadsheet per curve extracted and, in each spreadsheet, as many rows as in the ExperimentDesign file. We provide sample c3d files with simulated data in the folder called MyData.

An overview of the whole process is provided in the figure on the right hand side.



## Statistical analysis

The statistical analysis requires R [2]. R is an open-source statistical software that includes a scripting language and a large, and expanding, number of libraries to perform all sorts of computations. Had it been possible, we would have also developed the data extraction script with R. However, the biomechanical toolbox is not compatible with R. Note that the R script may use any correctly formatted XL file, not necessarily created with the Matlab script described above.

For those new to R, the easiest route is first to download and install R and download and install Rstudio, an open-source integrated development environment. The following sections detail how to run the EstVcomp.R script to obtain the variance components as per the description in [3].

### Required libraries:

1. lme4
2. latticeExtra
3. grid
4. readxl
5. tidyr

If these R libraries are not installed, you can install them by typing the following R snippet:

```
install.packages(c("lme4","latticeExtra","grid","readxl","tidyr"))
```

### User-specified parameters

The first 2 lines in the R script asks for 2 parameters:

1. *specfile* – the full path to the excel file which contains the specification of the experiment. It should have the same format as the example file ‘ExperimentDesign.xlsx’. For detail see section ‘Specifying Design.’
2. *formstr* – the formula which specifies the linear mixed model that you want to fit. It needs to reflect to relationship between the different sources of variance components in the experiment. If you are familiar with the R package lme4, then you probably know how to specify the formula, otherwise, see the Section ‘Specifying formstr’ below.

### Specifying Design

The design file should be an excel spreadsheet with 3 sheets:

- Help
- Parameters
  - It should contain 3 columns: Trajectory is the joint angle, Dim is the dimension, and Output file name is the *full file path* of the file that stores the data.
  - Note that only the variables implied by the Trajectory and Dim columns will be included in the statistical analysis, meaning the Data excel file could contain many extra sheets of information that will be ignored.
- Experiment design
  - This R script has been tested with the column provided. However, different experimental design could be catered for, just add as many columns as you need and adjust the R script and formula (cf. below).
  - SideName denotes the side. Note that data for different sides are analysed separately, so difference in sides are not considered variance components. If there are no sides (e.g. the data is the averaged of both sides), then just put one of ‘L’ or ‘R’.

- We assume there is always one (and one only) stride per trial. So the column StrideNumber is ignored by the variance component analysis (but used by the data extraction Matlab script)

## Specifying *formstr*

The parameter formstr is always of the form '~(1|A)+(1|B)+…'. That is, it starts with a tilde '~', and then adds terms of the form `(1|X)`. Here `X` denotes the variance components. And they have to match the names/spelling as those specified in the headers of the Experiment design file.

Note that, if something is not varying, it will not contribute to the variability. For example, if there is only one Assessor involved in the experiment, the term for Assessor should not appear in the formula.

Finally, the term corresponding to the innermost source, typically the inter-trial variability, need not be specified, as it is the same term as the residual term.

## Interaction

If we want to fit the interaction effect between components A and B, then we use '(1|A:B)'. The order of the factors are irrelevant, so `(1|A:B)` and `(1|B:A)` give the same result.

## Crossed effects

When we say components A and B are *crossed*, we need to fit both the main effects '(1|A)`, `(1|B)` as well as their interaction effect '(1|A:B)'.

## Nested effects

When components A is nested within B, then we can use the term '(1|B/A)'. An equivalent notation is to fit the main effect of the parent component, plus the interaction term. That is, the above term expands to '(1|B)+(1|B:A)`

## Examples formulae

**The example of the article:** Multiple subjects, multiple assessors, within each subject-assessors combination, there are multiple sessions, within each session, there are multiple trials.
~(1|Subject)+(1|Assessor)+(1|Subject:Assessor)+(1|Subject:Assessor:Session)

One subject, multiple assessors, multiple days, within each day, multiple sessions, with each session, multiple trials.
~(1|Assessor)+(1|Day)+(1|Assessor:Day)+(1|Assessor:Day:Session)

One subject, multiple assessors, one session, within each session, multiple trials.
~(1|Assessor)+(1|Assessor:Session), or simply ~(1|Assessor/Session)

Multiple subjects, multiple assessors, within each subject-assessors combination, one session, and within each session, multiple trials.
~(1|Subject)+(1|Assessor)+(1|Subject:Assessor)

Additional help with formula may be found here:
https://au.mathworks.com/help/stats/linearmixedmodel-class.html#btxc3mh.

## The main function

The main function in the script is estVcomp. It takes two inputs, which correspond to two potential outputs:

- *savePlot* – either TRUE or FALSE. If TRUE the final plot is saved in a pdf in the working directory.
- *saveSD* – either TRUE or FALSE. If TRUE the datasets of estimated standard deviations is saved as a .csv in the current directory.

### Renaming variance components

When the function is executed, it will list the default names of the variance components. And it will ask whether you want to rename them. If you answer 'y', you will be further prompted to enter the new name for each variance component in turn. When you are entering the new name, do not use quotation mark. Just type the name.

### Interim plot

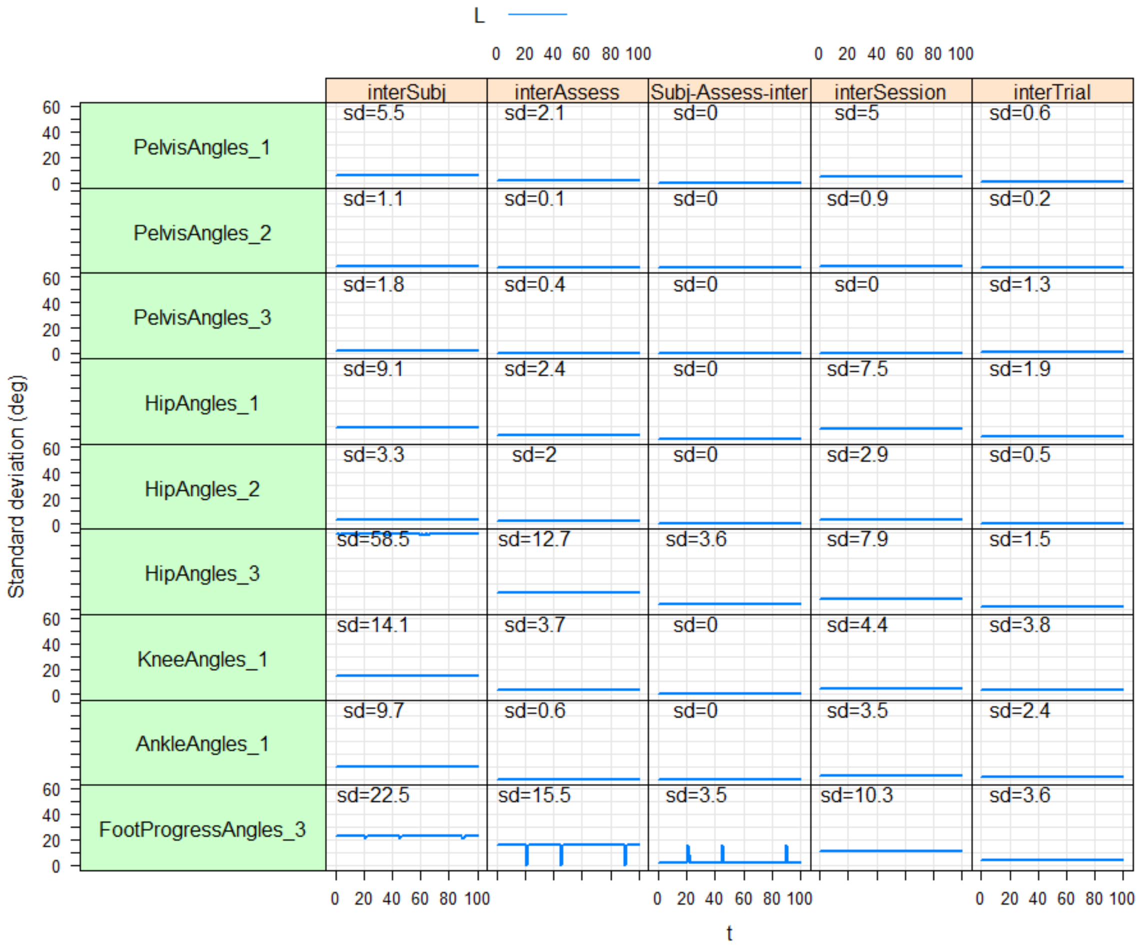
During the analysis, the function will plot the variance component curves for the particular subset of data being analysed. But only the final plot is saved (if savePlot=TRUE).

# Example Run

The default parameters are set to work with the example data in ExperimentResults.xls and ExperimentDesign.xls. If you execute the entire Rscript, it will define the parameter, define the function estVcomp and run it.

Using the example data, the final plot should look like the one below.
Things to note:

1. The SD curves with the example files are all flat line. This is purely due to the fact that the example data were simulated by adding random constants to the entire curve. Therefore, the variability stays the same throughout the timeline. But in general, this would not happen.
2. The legend only has 'L' with a blue line, this is because there is no data for the right-hand-side in the example dataset. Otherwise, the legend of 'R' should also appear.
3. The number in the top left corner of each panel is the average SD for that panel.
4. The column headings are the variance component names. The default names are usually quite long and can get squashed, so it is almost always better to rename them when prompted. Suggested renames are:
   - SubjectName → interSubject
   - AssessorName → interAssessor
   - SubjectName:AssessorName → Subj-Assess-inter
   - SubjectName:AssessorName:SessionName → intersession
   - Residual → interTrial
5. The inter-Subject and Subject-Assessor-interaction variability are not reported in [3] because the data for Subject variability is not available.
6. The row names are the names of the variables we are measuring.
7. Note that if irregularities in the curve appear, for example, in the FootProgressAngles_3-AssessorName panel, where we see some estimated SD dropping to 0. This is indication that the underlying mixed model failed to converge. Typically, the R script will return the warnings which you can see by running `warnings()`. If this occur, it will be worth investigating those specific data. But usually, the overall trend should be clear that the final plot itself will indicate what the correct value should be.



## References

1. Barre A, Armand S. Biomechanical ToolKit: Open-source framework to visualize and process biomechanical data. Computer methods and programs in biomedicine. 2014;114(1):80-7.

2. {R Core Team}, R: A language and Environment for Statistical Computing., (2015).

3. Chia K, Sangeux M. Quantifying sources of variability in gait analysis. Gait & Posture, *In press.*