Bioconductor's stepNorm package

Yuanyuan Xiao[1] and Yee Hwa Yang[2]

October 18, 2004

Departments of [1]Biopharmaceutical Sciences and [2]Medicine
University of California, San Francisco
yxiao@itsa.ucsf.edu
jean@biostat.ucsf.edu

Contents

1 Overview

This document provides a tutorial for the `stepNorm` package, which performs a stepwise within-slide normalization procedure STEPNORM on two-channel cDNA spotted arrays. Two-channel microarrays measure relative abundance of expression of thousands of genes in two mRNA populations. This relative abundance is usually expressed as ratios, $M = log_2 \frac{R}{G}$, where R and G are the fluorescent intensity measurements of the red and green channels. The most pronounced systematic variation embodied in the ratios that does not contribute to differential expression between the two mRNA populations is the imbalance of the green and red dye incorporation. This imbalance is manifested as the dependence of ratios on primarily two factors, the fluorescent intensity (A) and the spatial heterogeneity (S).

STEPNORM is a normalization framework that integrates various models of different complexities to sequentially detect and adjust systematic variations associated with spot intensities (A), print-tips (PT), plates (PL) and two-dimensional spatial effects. For more details on STEPNORM, the

reader is referred to Xiao et al. (2004).

Functionalities in `stepNorm`. The `stepNorm` package implements the STEPNORM procedure which is based on a series of robust adaptive location normalization methods correcting for different types of dye biases (e.g. intensity, spatial, plate biases). It enables the user to perform normalization either in a single-step or a sequential fashion. Further, it allows the use of control sequences spotted onto the array and possibly spiked into the mRNA samples.

Microarray classes. The `stepNorm` packages relies on microarray class definitions in `marrayClasses`. You should also install this package and consult its vignette for more information.

Case study. We demonstrate the functionality of the `stepNorm` package using a *swirl* zebrafish slide. The swirl experiment is comprised of four replicate hybridizations that contain 8,448 spots. It was carried out using zebrafish as a model organism to study the effect of a point mutation in the BMP2 gene that affects early development in vertebrates (Yang et al. (2002)).

Help files. As with any R package, detailed information on functions, classes and methods can be obtained in the help files. For instance, to view the help file for the function `stepWithinNorm` in a browser, use `help.start()` followed by `?stepWithinNorm`.

2 Case study : The *swirl* Experiment

2.1 Data

We demonstrate the functionality of this package using gene expression data from the *swirl* experiment. Two sets of dye-swap experiments were performed, for a total of four replicate hybridizations. For each of these hybridizations, target cDNA from the swirl mutant was labeled using one of the Cy3 or Cy5 dyes and the target cDNA wild-type mutant was labeled using the other dye. Target cDNA was hybridized to microarrays containing 8,448 cDNA probes, including 768 controls spots (e.g. negative, positive, and normalization controls spots). Microarrays were printed using 4 times 4 print-tips and are thus partitioned into a 4 times 4 grid matrix. Each grid consists of a 22 times 24 spot matrix that was printed with a single print-tip. Here, spot row and plate coordinates should coincide, as each row of spots corresponds to probe sequences from the same 384 well-plate. Raw images of the Cy3 and Cy5 fluorescence intensities for all four hybridizations are available at `http://fgl.lsa.berkeley.edu/Swirl/index.html`. To load the dataset, use `data(swirl)`, and to view a description of the experiements and data, type `?swirl`.

```
> library(stepNorm)
```

```
Loading required package: MASS
Loading required package: marray
```

```
> data(swirl)
> maNsamples(swirl)
```

```
[1] 4
```

```
> maNspots(swirl)
```

```
[1] 8448
```

2.2 Single-step Normalization

The `stepNorm` package provides the function `withinNorm` to conduct normalization in a single-step fashion. For instance, The following commands applies the scatter plot smoother *loess* within each print-tip-group on a *swirl* slide.

```
> lpt.swirl <- withinNorm(swirl[, 1], norm = "loessPrintTip")
```

The function `withinNorm` is a simple wrapper function provided for users interested in conducting a set of standard normalization methods with default parameters (though supplying user desired parameters is also feasible); its functionalites will be elaborated in the next section.

2.3 Stepwise Normalization with a Model Selection Component

As biases are slide- and experiment-dependent, different slides may show different intensity and spatial trends. Using one model (step) to correct all biases in a slide or using the same model for different slides exhibiting different biases might not be adequate. The function `stepWithinNorm` implements the normalization procedure STEPNORM, which integrates a number of models under the same framework and assesses their effectiveness via a quantitative criterion. Such a process is applied to each individual slide in an experiment so that data (slide) specificity could be achieved. Unlike single-step normalization methods, STEPNORM could avoid data under-fitting or over-fitting as it implements both bias detection and removal in the same context. The following command using the function `stepWithinNorm` applies a default stepwise procedure, which adjusts A, PT, PL and Spatial biases in an ordered succession, on a *swirl* slide. Appropriately, the STEPNORM procedure chooses *loess*, median shift and median shift for the correction of the A, PT and PL biases respectively and deems spatial heterogeneity on the slide not significant enough to warrant adjustments. Diagnostic plots before and after each step of bias adjustment are shown in Figure 1.

```
> step.swirl1 <- stepWithinNorm(swirl[, 1])

Normalizing slide  1 ...

BIC of null model:  -5328.85

step  1 -- wholeChipA :
BIC of methods ( med rlm loess ) are:  -10138.86 -11389.80 -12019.94
chosen :  loess

step  2 -- printTipA :
BIC of methods ( med rlm loess ) are:  -12278.60 -12120.52 -11695.52
chosen :  med

step  3 -- plateA :
BIC of methods ( med rlm loess ) are:  -12821.92 -12750.23 -12250.72
chosen :  med
```

```
step  4 -- wholeChipSpatial :
BIC of methods ( rlm2D loess2D aov2D spatialMed ) are:  -12789.12 -12674.70 -12212.33 -12657.82
this normalization step is not necessary

Slide 1 normalization steps:  wholeChipA-loess-> printTipA-med-> plateA-med

> norm.swirl <- step.swirl1[[1]]
> step.swirl1[[2]]

[[1]]
                 From    To  Deviance   Enp Penalty Criterion
null                         -5328.85  0.00          -5328.85
wholeChipA            loess -12064.00 +4.87    9.04 -12019.94
printTipA               med -12467.33   +16    9.04 -12278.60
plateA                  med -13209.57   +22    9.04 -12821.92
wholeChipSpatial            -13209.57 +0.00         -12821.92
```

2.4 Sequential Normalization.

In addition to the `stepWithinNorm` which includes a model selection component, the `stepNorm` package also provides a multi-step normalization function `seqWithinNorm`, which conducts normalization in a user specified sequence without the heavy computation burden of choosing among models. For instance, the following command employs "`loess`" for correction of the A bias, and the global median shift ("`median`") for the PT and PL biases and no normalization ("`none`") on the spatial bias.

```
> step.swirl1 <- seqWithinNorm(swirl[, 1], A = "loess", PT = "median",
+     PL = "median", Spatial2D = "none")

Normalizing slide  1 ...

Deviance of null model: -5328.85
enp          Deviance          BIC
4.87          -12064          -12019.94
16           -12467.33          -12278.6
22           -13209.57          -12821.92
0           -13209.57          -12821.92
```

3 The `withinNorm` function

The function `withinNorm` provides a series of single-step standard normalization. It wraps around functions `fitWithin` and `fit2DWithin` and returns an object of class `marrayNorm`. It has three arguments

`marraySet`: Object of class `marrayRaw` and `marrayNorm` containing intensity data for the batch of arrays to be normalized.

`subset`: A logical or numeric vector indicating the subset of points used to fit the normalization model.

`norm`: Character string specifying the normalization method. Thirteen normalization procedures are available with this function:
`none`, no normalization;
`median`, global median location normalization;
`rlm`, global intensity or A–dependent robust linear normalization;
`loess`, global intensity or A–dependent robust nonlinear normalization;
`medianPrintTip`, within-print-tip-group median normalization;
`rlmPrintTip`, within-print-tip-group *rlm* normalization;
`loessPrintTip`, within-print-tip-group *loess* normalization;
`medianPlate`, within-well-plate-group median normalization;
`rlmPlate`, within-well-plate-group *rlm* normalization;
`loessPlate`, within-well-plate-group *loess* normalization;
`aov2D`, spatial bivariate location normalization using ANOVA (Sellers et al. (2003));
`rlm2D`, spatial bivariate location normalization using the `rlm` function;
`loess2D`, spatial bivariate location normalization using the `loess` function;
`spatialMedian`, spatial location normalization using a spatial median approach (Wilson et al. (2003)).

. . . : Misc arguements for the specified 'norm' function

The function `withinNorm` is simple to use, the user needs only to input the data as a `marrayRaw` (or `marrayNorm`) object and indicate the normalization method intended as a character string. It is also flexible to change parameters. For example the default *loess* normalization procedure uses $span = 0.4$, if a smaller span is desired, it can be specified as follows,

```
> lpt.swirl <- withinNorm(swirl[, 1], norm = "loess", span = 0.2)
```

The user should consult the functions `fitWithin` and `fit2DWithin` (using `?fitWithin` and `?fit2DWithin`) for details on extra parameters suited for each normalization method.

4 The `stepWithinNorm` function

The `stepWithinNorm` is the main function that carries out the STEPNORM procedure which adjusts biases sequentially and in a stepwise normalization. In each step one bias is targeted for correction. Figure **??** illustrates default steps by `stepWithinNorm`, which follows the successive correction of intensity (A), print-tip (PT), plate (PL) and spatial heterogeneity (2D Spatial) biases. Within each step several models are employed competitively and the model that achieves the best balance between the goodness of fit and simplicity is chosen for application before proceeding to the next step. The `stepNorm` pacakge implement two model selection criteria, the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC). We recommend using the latter as a typical microarray data consists of tens of thousands spots, and penalty in BIC for number of model complexity is more appropriate for such big datasets.

The function `stepWithinNorm` has four arguments (see also `?stepWitinNorm`):

`marrySet`: Object of class `marrayRaw`or codemarrayNorm, containing intensity data for the batch of arrays to be normalized.

`subset`: A "logical" or "numeric" vector indicating the subset of points used to compute the normalization values.

`wf.loc`: A list, each component of which is a step for the removal of a particular systematic varation. Typically each step is also a list of several candidate models of different complexity, such models can be specified using functions `fitWithin` and `fit2DWithin` (see `?fitWithin` and `?fit2DWithin`). The function `makeStepList` is also provided for a user friendly approach to construct such a list; the user only needs to input the model names as character strings, see `?makeStepList` If missing, the default procedure will be used, which we consider appropriate for most slides (for details about the default procedure, see `?stepWithinNorm`).

`criterion`: Character string specifying the criterion used for the selection of the best normalization procedure in each step. Choices include ``BIC'' and ``AIC''; if no specification is made, the default is ``BIC''.

Normalization is performed simultaneously for each array in the batch using the same stepwise procedure. We illustrate next on specifying customized stepwise procedure using the function `makeStepList`. There may be cases when model selection for a certain step is not necessary, for example, when the nonlinear trend between M and A is evident, the user may wish to apply *loess* directly on the slide rather than enduring unnecessary computation on model comparisons. In addition, the user desires no normalization on the Spatial 2D step. Such a procedure can be specified conveniently using the function `makeStepList`:

```
> wf.loc <- makeStepList(A = "loess", Spatial2D = NULL)
> step.swirl1 <- stepWithinNorm(swirl[, 1], wf.loc = wf.loc)

Normalizing slide  1 ...

BIC of null model:  -5328.85

step  1 -- WholeChipA :
BIC of methods ( loess ) are:  -12019.94
chosen :  loess

step  2 -- PrintTipA :
BIC of methods ( median rlm loess ) are:  -12278.60 -12120.52 -11695.52
chosen :  median

step  3 -- PlateA :
BIC of methods ( median rlm loess ) are:  -12821.92 -12750.23 -12250.72
chosen :  median

Slide 1 normalization steps:  WholeChipA-loess-> PrintTipA-median-> PlateA-median
```

However, `makeStepList` uses defalut parameters, for example, the span of *loess* is set at 0.4 for the correction of the A bias. To employ parameters other than the defaults, the user needs to construct the `wf.loc` list directly as the follows:

```
> A <- list(loess = fitWithin(fun = "loessfit", span = 0.2))
> PT <- list(med = fitWithin(z.fun = "maPrintTip", fun = "medfit"),
+     rlm = fitWithin(z.fun = "maPrintTip", fun = "rlmfit"), loess = fitWithin(z.fun = "maPrintT
+         fun = "loessfit"))
> PL <- list(med = fitWithin(z.fun = "maCompPlate", fun = "medfit"),
+     rlm = fitWithin(z.fun = "maCompPlate", fun = "rlmfit"), loess = fitWithin(z.fun = "maCompP
+         fun = "loessfit"))
> wf.loc <- list(wholeChipA = A, PrintTip = PT, Plate = PL)
> step.swirl1 <- stepWithinNorm(swirl[, 1], wf.loc = wf.loc)
```

5 The seqWithinNorm function

The seqWithinNorm function allows the user to conduct a sequential normalization procedure without the heavy computational overhead of model selection. The user can specify an appropriate model in each step or even skip a step.

The function stepWithinNorm has four arguments (see also ?stepWitinNorm):

marrySet: Object of class marrayRaw or marrayNorm, containing intensity data for the batch of arrays to be normalized.

subset: A "logical" or "numeric" vector indicating the subset of points used to fit the normalization models.

loss.fun: The loss function used in calucating deviance, the default uses squared sum of residuals; for absolute sum of residuals, use abs.

A: A character string specifying the normalization method for the adjustment of intensity or A bias; choices include ''median'', ''rlm'', ''loess'' and ''none''. The default is set as ''loess''.

PT: A character string specifying the normalization method for the adjustment of print-tip or PT bias; choices include ''median'', ''rlm'', ''loess'' and ''none''. The default is set as ''median''.

PL: A character string specifying the normalization method for the adjustment of well-plate or PL bias; choices include ''median'', ''rlm'', ''loess'' and ''none''. The default is set as ''median''.

Spatial2D: A character string specifying the normalization method for the adjustment of spatial 2D bias; choices include ''rlm2D'', ''aov2D'', ''loess2D'', ''spatialMedian'' and ''none''. The default is set as ''none'.

criterion: Character string specifying the criterion, ''AIC'' or ''BIC''. If no specification, ''BIC'' is used. Note that here criterion is calculated solely for informaion purpose.

To conduct *loess* normalization for the A step, median normalization for the PT step, and no further normalization for the rest steps, the user can use the command below,

```
> step.swirl1 <- seqWithinNorm(swirl[, 1], A = "loess", PT = "median",
+     PL = "none", Spatial2D = "none")
```

```
Normalizing slide  1 ...

Deviance of null model: -5328.85
enp          Deviance          BIC
4.87          -12064          -12019.94
16           -12467.33         -12278.6
0           -12467.33          -12278.6
0           -12467.33          -12278.6
```

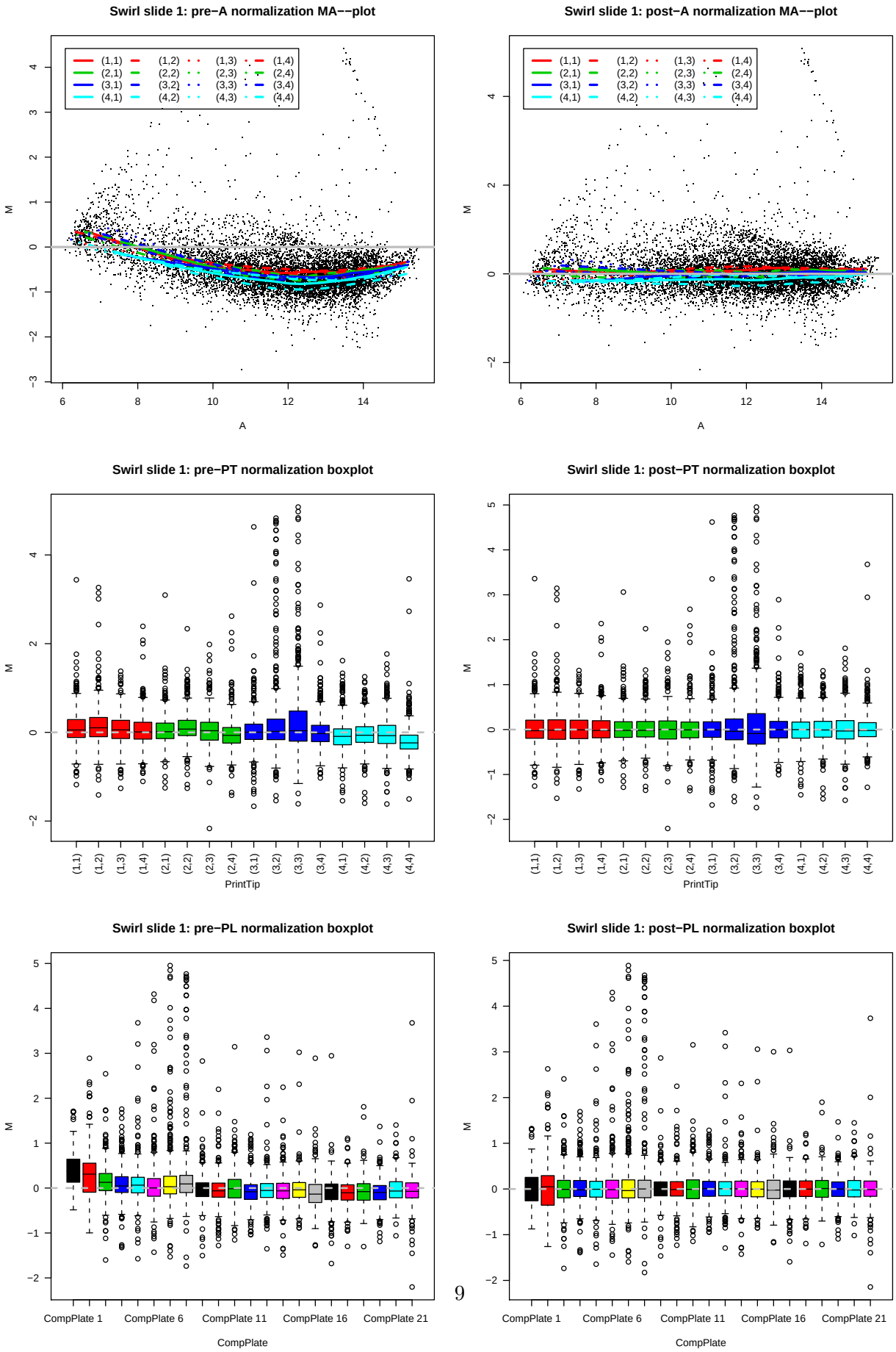
References

K. F. Sellers, J. Miecznikowski, and W. F. Eddy. Removal of systematic variation in genetic microarray data. Technical report, Department of Statistics, Carnegie Mellon University, 2003.

D. L. Wilson, M. J. Buckley, C. A. Helliwell, and I. W. Wilson. New normalization methods for cdna microarray data. *Bioinformatics*, 19:1325–1332, 2003.

Y. H. Yang, S. Dudoit, P. Luu, D. M. Lin, V. Peng, J. Ngai, and T. P. Speed. Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Research*, 30(4):e15, 2002.

Y. Xiao, M. R. Segal, and Y. H. Yang. Stepwise normalization of two-channel cDNA microarrays. *manuscript in preparation*.


Swirl slide 1: pre−A normalization MA−−plot
Swirl slide 1: post−A normalization MA−−plot
(1,1) (1,2) (1,3) (1,4)
(2,1) (2,2) (2,3) (2,4)
(3,1) (3,2) (3,3) (3,4)
(4,1) (4,2) (4,3) (4,4)
M
A
Swirl slide 1: pre−PT normalization boxplot
Swirl slide 1: post−PT normalization boxplot
PrintTip
Swirl slide 1: pre−PL normalization boxplot
Swirl slide 1: post−PL normalization boxplot
CompPlate 1
CompPlate 6
CompPlate 11
CompPlate 16
CompPlate 21
CompPlate

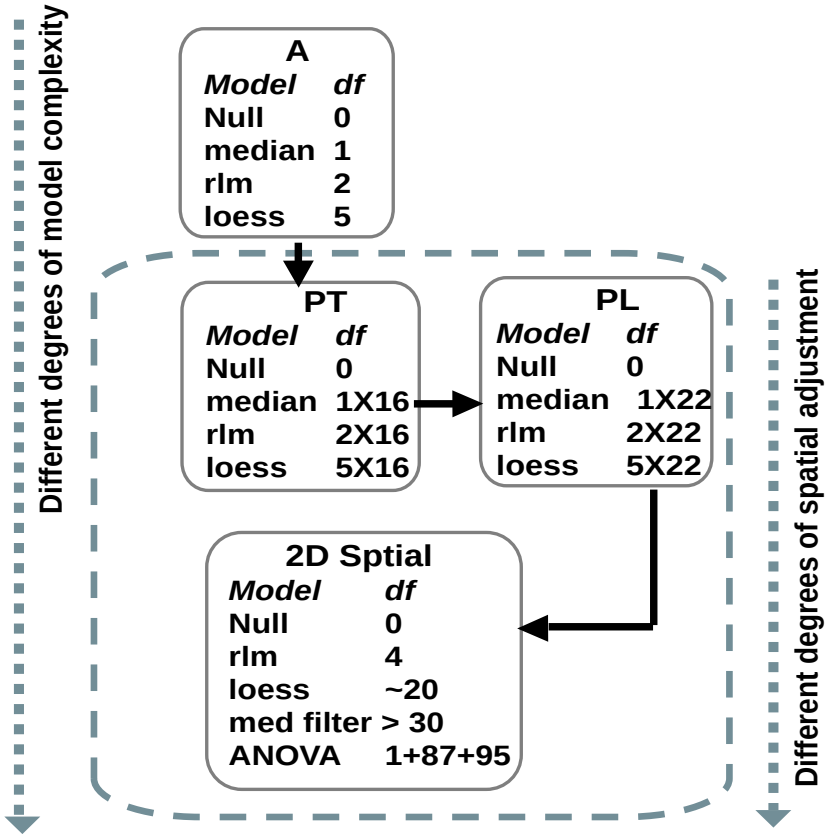


Figure 2: STEPNORM procedures for the swirl experiment. The swirl slides have 16 print-tips, 22 well plates, 88 rows and 96 columns.