NAME

maf − minimum/maximum autocorrelation factors

SYNOPSIS

maf [**−maf** [*nshifts* [*sd1* [*sr1* [*sd2* [*sr2* [*sd3* [*sr3* [*sd4* [*sr4*]]]]]]]]]
| **−mnf** | **−prc** | **−fac** | **−cd**
| **−can** [*nset1*] | **−cann** [*nsets* [*nset1 ... nsetn*]] | **−mad** [*nset1*]] | **−pls** [*nset1*]]
[**−corr**]
[**−eq**]
[**−varimax** | **−quartimax]**
[**−mineigen** [*minimum_eigenvalue*] | **−nfact** *nfactors*] [**−com**]
[**−mean** [*size*] | **−median** [*size*] | **−sar** | **−N** *dispfile*]
[**−gauss** [*rho* [*trunc*]]]
[**−M** *maskfile* [*maskvalue*]]
[**−C** *dispfile*]
[**−WC** *weightCfile*]
[**−WN** *weightNfile*]
[**−saveC** *saveCfile*]
[**−saveN** *saveNfile]*
[**−savecorr** *savecorrfile]*
[**−s** [*rows* [*cols*]]]
[**−p** [*frow* [*fcol*]]]
[**−saveT** *saveTfile*]
[**−log** *logfile*]
[**−nolog**]
[**−eigen** [*size*]

DESCRIPTION

maf performs a linear transform on an image sequence with the purpose of isolating the signal from the noise. The input sequence must be in HIPS-format and the format must be either byte, short, int or float. The output sequence is in float. The inputsequence may be either bandinterleaved by line or not. *maf* also supports the IMM defined irregular format. The extended byte parameter 'Irregular' which may have the values one, two or three, indicates that either one, two or three position variables should be read prior to the data. Statistical data are sent to a file whose name defaults to 'log(pid)'.

OPTIONS

−maf [*nshifts* [*sd1* [*sr1* [*sd2* [*sr2* [*sd3* [*sr3* [*sd4* [*sr4*]]]]]]]]] Calculate minimum/maximum autocorrelation factors using a pool of *nshifts* difference covariance matrices, estimated using the specified shifts (*sd* , *sr*). The default is a pool of one horizontal shift (0,1) and one vertical shift (1,0).

−mnf Calculate the maximum noise fraction transform. If **−mean** *[size]* or **−median** [*size*] is specified the noise is estimated as the difference to the local mean respectively median. In the case of median filtering the default is a square filter, a rotation symmetric filter can be specified using option **-rotsym** In the case of mean filtering the filters are rotation symmetric and the kernelsize is *size.* The default kernelsize is 3 times 3 pixels. If **−gauss** [*rho*[*trunc*]] is specified, the mean or median filter will be with gaussian weights. *rho* is the standard deviation of the Gaussian weights. *trunc* is the ratio of the largest and the smallest weight. If *rho* and *trunc* are not specified the values *rho* = 1 *trunc* = 0.05 will be used. If **−sar** is specified a the noise is estimated as the residual in a simultaneous autoregressive proces using the W, NE, N, NW neighbours. Finally if the option **−N** *dispfile* is used the covariance structure of the noise is read from *dispfile.* The covariance matrix must be loaded in HIPS-format and the format must be double. The first row could contain the noise mean, but is not used. The default is to estimate the noise as the difference between the image and a median filtered (3 times 3 pixels) version of the image.

−prc Calculate principal components.
If **−corr** is specified the correlation matrix of the data is used instead of the covariance matrix. **−eq** use equal weifgts of the mean when calculating the among groups matrix in the canonical

discriminant analysis.

−can [*nset1*]
Calculate canonical correlations between the *nset1* first frames and all other frames. The first *nset1* output frames contains linear combinations of set 1, and the following *nset2* output frames contains linear combinations of set 2. The 2 times min(*nset1* , *nset2*) canonical variates are found as the first frames within the linear combinations of each set. For the largest set the remaining frames contain the residual information not accounted for by the canonical variates. *nset1* default to one half of the frames.

−pls [*nset1*]
Calculate the partial least squares transformation between the *nset1* first frames and all other frames. The first *nset1* output frames contains linear combinations of set 1, and the following *nset2* output frames contains linear combinations of set 2. The 2 times min(*nset1* , *nset2*) variates are found as the first frames within the linear combinations of each set. For the largest set the remaining frames contain the residual information not accounted for by the canonical variates. *nset1* default to one half of the frames.

−cann [*nsets* [*nset1* ... *nsetn*]]
Calculate the canonical correlations between more than two sets. The canonical variables are sent to standard output. If only the number of sets *nsets* is specified, it is assumed that all sets have equal number of variables.

−mad [*nset1*]
Calculate multivariate alteration detection factors using the *nset1* first frames as set 1 and all others as set 2. *nset1* default to one half of the frames.

−fac Perform principal factor analysis. the number of factors is determined either by specifying **−nfact** *nfactors* where *nfactors* is the requested number or factors, or by specifying **−mineigen** *minimum_eigenvalue* i.e. using the mineigen criterion (only factors corresponding to eigenvalues greater than *minimum_eigenvalue* are calculated. The minimum eigenvalue is 1 by default). To use the communalities estimate on the diagonal of the correlation matrix use **−com** else ones will be used on the diagonal. Either varimax or quartimax rotation or none at all can be specified. This is done by use of **−varimax** or **−quartimax.**

−cd perform canonical discriminant analysis. Either a training set or a set of covariance matrices must be specified. The training set should be a HIPS byte image with classes numbered 1, 2, 3 ... . Pixels that are not part of the training set should have the value 0. The training set is specified using the **−M** *maskfile* option. A set of covariance matrices should be a sequence of HIPS images. In the first row the images should have the class mean, in the first column of the second row the images should have the number of pixels used in the estimations for this class, and in the following rows of each image the covariance matrix for the class should be. The first image in the sequence should contain the statistics for all classes taken as one. The set of covariance matrices may be stored in a separate file by use of the option **−saveC** *cov.* Note that the squared canonical correlations given in the logfile are the percentage of the variation in the training set in the respective canonical variable that is explained by the model, i.e. SAK_mod/SAK_tot. The eigenvalues are SAK_mod/SAK_res.

−nfact *nfactors*
Calculate only *nfactors* factors and send these to <stdout>. The default is that all are sent.

−C filename
Read the covariance matrix of the input data from *filename* The covariance matrix must be loaded in HIPS-format and the format must be double. The first line must contain the image mean.

−WC filename
Read a weight image to be used for the estimation of the covariance matrix from *filename*.

−WN filename
Read a weight image to be used for the estimation of the noise covariance matrix from *filename*.

−s [*rows* [*cols*]]
Only an extracted image of size *rows* times *cols* is sent to output. *rows* default to one half of the image size *cols* default to *rows*. **−p** [*frow* [*fcol*]] The extracted output starts at pixel *frow* , *fcol*. *frow* and *fcol* default so that the extracted section is centered.

−M *maskfile* [*maskvalue*]
Read a mask from *maskfile*. The mask must be HIPS-format and the format must be byte. **maf** will only calculate statistics on those pixels in the input sequence whose value in *maskfile* is *maskvalue*.

−saveC *saveCfile*
Save the estimated mean and covariancematrix in *saveCfile*.

−saveN *saveNfile*
Save the estimated mean and covariancematrix of the noise in *saveNfile*.

−savecorr *savecorrfile*
Save the eigenvalues and the estimated correlations between the bands and the factors in *savecorrfile*. The eigenvalues will be in the first row.

−saveT *saveTfile*
save the transformation matrix in file for later use for instance in the program **lintrans** to transform back to the original space. The transformation matrix is stored in *saveTfile*. The first row contains the mean values of the original variables.

−log *logfile*
use the name **logfile** for the logfile instead of log'pid'.

−nolog suppress logfile

−eigen [*size*]
perform an eigenvalue filter analysis. The input must be a single image (Irregular format is not allowed). Features are generated by translating the image over a local neighbourhood, the size of which is given by *size* . This option may be used with principal components analysis (default) as well as canonical discriminant analysis.

SEE ALSO
disc(1), disc3D(1), seed(1), bil(1), roprc(1), lintrans(1), musecc(1)

AUTHOR
Rasmus Larsen
Department of Mathematical Modelling, Technical University of Denmark
e-mail: rl@imm.dtu.dk

ACKNOWLEDGEMENTS
An earlier version of this program was developed by the author as a student under the supervision of Professor Knut Conradsen and Associate Professors Dr. Bjarne Ersboell and Dr. Allan Aasbjerg Nielsen. In particular, Dr. Nielsen has suggested a number of the implemented transformations and statictical outputs and executed extensive testing of the program.

REFERENCES
Green et al.: A transformation for Ordering Multispectral Data in Terms of Image Quality with Implications for Noise Removal. *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 26. No. 1, pp. 65−74, 1988

Switzer & Green: Min/max Autocorrelation Factors for Multivariate Spatial Imagery. *Tech. rep. No. 6, Department of Statistics, Stanford University, 10 pp., 1984.*

Conradsen, Knut: En Introduktion til Statistik, bind 2B. *IMSOR/DTH, 1992*

Nielsen, Allan Aasbjerg & Conradsen, Knut: Multivariate Alteration Detection (MAD) In Multispectral Bi-temporal Image Data: A New Approach to Change Detection Studies. Department of Mathematical Modelling, technical University of Denmark, Technical Report IMM-REP-1997-11

Kettenring, J.R.: Canonical Analysis of Several Sets of Variables. *Biometrika, Vol. 58, 3, p.433*