A Source Separation Model for Eroded Soils in Uplands

(22S:138, Fall 2005)

By Ozan Abaci & Pradeep Mandapaka Venkata

ABSTRACT: Clear Creek, a tributary of the Iowa River, is located within Iowa and Johnson counties and has been a target watershed by the State's Geological Survey, IDNR, NRCS and other non-profit organizations (e.g. Clear Watershed Board). Different anthropogenic activities and natural events (e.g., farming, floods) in the Clear Creek watershed over the last fifty years have contributed to the increased influx of fine sediments and pollutants into the Creek (Langel, 1996). In order to be able to take the necessary precautions, it is important to understand the source of the suspended sediment which is the eroded soil delivered into the river. A Bayesian un-mixing model might be a helpful tool to fraction mixtures of eroded-soils into their source contributions. Source soil is the potentially erodible soil from the upland and assuming no river bed contribution, suspended sediment is a mixture of the source soils. Due to the lack of measured data at this moment, data for both the source soil and the suspended mixture will be based on randomly created carbon, nitrogen and oxygen isotopes and the soil fertility ratio (C/N) from the Clear Creek Watershed, Iowa. The un-mixing model will be based on a previously written code by Jimmy Fox et al. and it will be modified into a more general form so that it will account for more suspended sediment sources and more isotopic tracers.

1. INTRODUCTION

Clear Creek, a tributary of the Iowa River, drains a watershed of approximately 103 mi^2, with 41 mi^2 being located in Iowa County. It takes its rise in Iowa County and divides the township in two parts in very nearly the middle. The Clear Creek Watershed starts near the town of Conroy and ends in Johnson County, at the Iowa River in Coralville. Clear Creek is significant in that it contains a watershed coalition group actively promoting positive land use practices and, as it spans a rural to urban gradient; it is a growing focus of efforts by the Iowa Department of Natural Resources. Whereas much current agroecological work is largely disciplinary, focused on plot to field scales, and is disconnected from Iowa's growing citizenry, improving Iowa's ecological and socioeconomic resilience requires a much broader view. (Rayburn A., Schulte L., and Merrick L. 2004) Since Iowa was first settled it has lost more than 95% of its wetlands, ranking third in the U.S. behind California and Ohio for percent of wetlands lost per state. (Lovetinsky L. 2001) Stream sinuosity has decreased over time, as wetlands have been drained, water has been canalized, and stream flow regimes have been altered. (Rayburn A., Schulte L., and Merrick L. 2004) As can be seen in the below figure, South Amana sub-catchment of the basin is prone to significantly higher erosion rates compared to other parts of the watershed. Unwarranted erosion presents a pressing problem for both upland land-users (e.g. farmers, logging industry) and in-stream fish habitats (Fox et al., 2002).

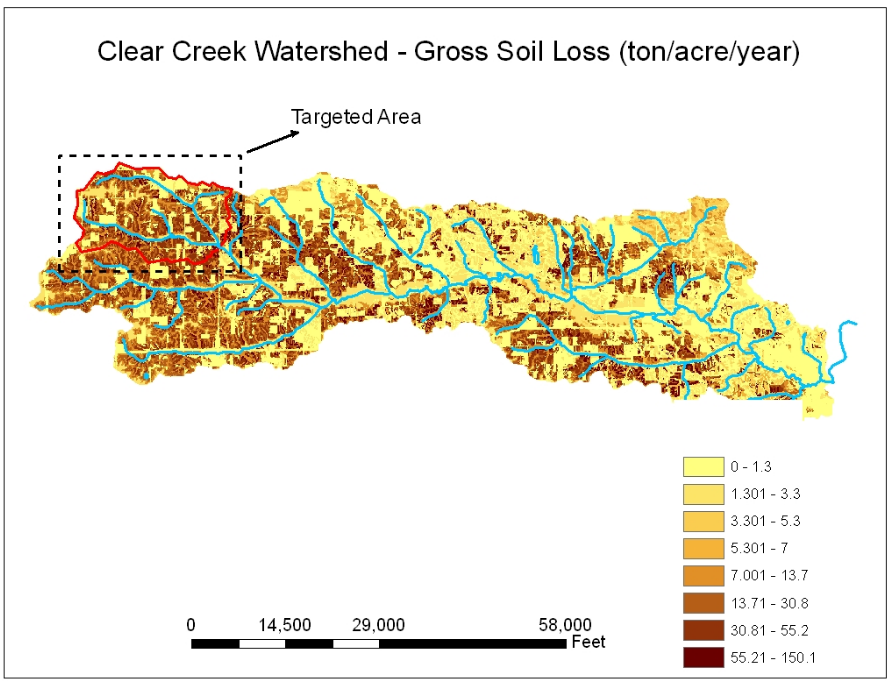

Figure 1 – Gross Soil Loss (tons/acre/year)

In order to develop more efficient and economically feasible erosion control practices, and to improve erosion prediction methods for better conservation planning we need to improve our understanding of soil erosion and sediment transport processes. Using stable isotopes as eroded soil tracers is a very dynamic and continuously evolving technique. Traditional techniques such as employing erosion pins and troughs to measure soil erosion from contrasting land-uses commonly face important spatial and temporal sampling constraints (Foster et al. 1990).

Isotopes are atoms of the same element that have the same numbers of protons and electrons but different numbers of neutrons. Isotopic compositions are determined in specialized laboratories using isotope ratio mass spectrometry. 4 stable isotopic ratios; Carbon, Nitrogen, Oxygen of Nitrate and C/N atomic ratio will be used to differentiate between soils that are prone to different land management practices and from different profile depths. $\delta^{15}N$, a soil property proportional to the ^{15}N:^{14}N isotopic ratio, has been used most prominently by environmental scientists to study nitrogen pollution in the hydrosphere and atmosphere; but $\delta^{15}N$ has also been used significantly for studying eco-environmental problems such as nitrogen cycling in vegetation and nitrogen cycling in lakes and coastal regions (Heaton, 1986; Lojen et al., 1997). $\delta^{13}C$, a soil property proportional to the ^{13}C:^{12}C isotopic ratio, highly retains the signature of the parent vegetation and therefore has been extensively used in soil research for paleo-environmental studies of changes in vegetation and climate and to investigate soil carbon dynamics (Balesdent and Mariotti, 1996; Keller et al., 2001). Oxygen isotope ratios of nitrate, which is proportional to the ^{18}O:^{16}O isotopic ratio, appear to be particularly useful, since they allow the differentiation between nitrate from atmospheric deposition ($\delta^{18}O_{Nitrate}$ between +25 and +70 ‰), nitrate from fertilizers ($\delta^{18}O_{Nitrate}$ between +23 ‰), and nitrate derived from nitrification processes in soils ($\delta^{18}O_{Nitrate}$ < 15 ‰). (Mayer et al., 2001) The C/N atomic ratio, defined as the ratio of total atomic carbon to total atomic nitrogen exists as a proxy typically used for ecosystem health and processes by soil fertility experts (Brady and Weil, 2001). Different source soils will have different ratios of above mentioned staple isotopes and they will contribute differently to the soil isotopic composition of the suspended mixture.

Possible factors such as season, profile depth, land management, etc is likely to change the soil's isotopic composition. As the possible soil isotopic composition changing factors, ***land use and profile depth*** are introduced into the un-mixing model. Physical role and contribution of each factor should be understood thoroughly before introducing them into the un-mixing model. For example, while profile depth and land use together form up 8 separate sources (See Table 1), including season as another factor (assuming every event ends within the same season that it occurs) requires as many separate runs as the number of season levels defined. There are 4 main land management practices currently being applied within the area of interest: Corn, soybean, forest and CRP (Conservation Reserve Program). Below figure shows different land management practices within the Clear Creek Watershed. A more detailed map is developed together with the USDA office at Williamsburg, Iowa and used for the sub-catchment of interest in order to distinguish between different management practices.

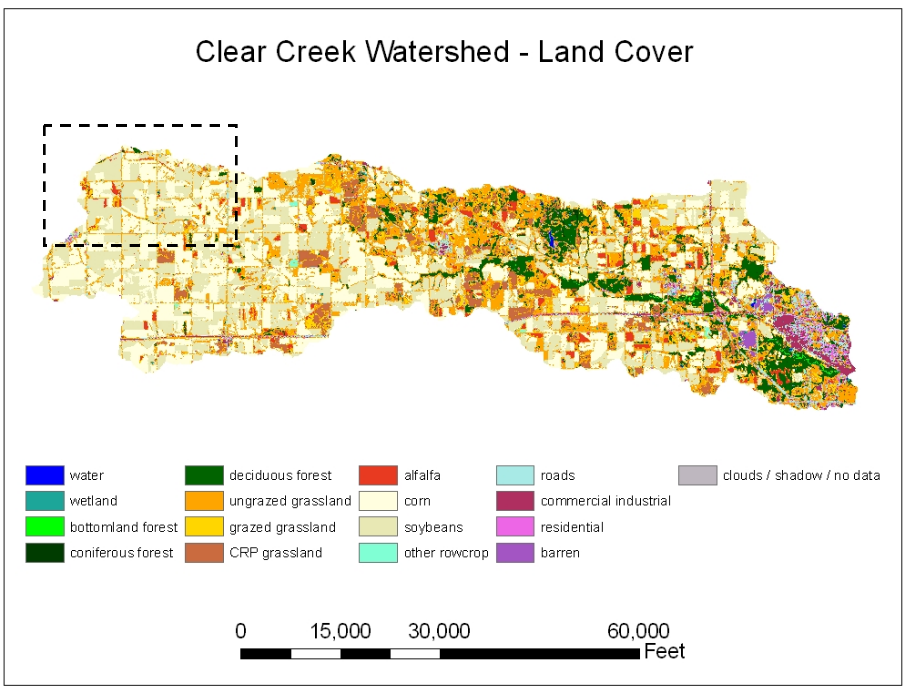

Figure 2 – Land Cover

Two different profile depths will be taken into account as no erosion is expected beyond the top 10 cm layer: 0~5cm and 5~10 cm from the surface. As the source data, soil samples need to be collected from each source and analyzed for carbon, nitrogen and oxygen isotopic ratios. Since there are 4 different land uses and 2 different profile depths taken into account, there will be 2 * 4 = 8 different soil sources contributing to the suspended sediment mixture that is being carried by the river flow. Suspended sediment mixtures should also be collected at the outlet of the watershed during strong rainfall events and analyzed for the tracer isotopes like the source soils.

Source Flag	Definition
1	Corn, 0~5 cm
2	Corn, 5~10 cm
3	Forest, 0~5 cm
4	Forest, 5~10 cm
5	Soybean, 0~5 cm
6	Soybean, 5~10 cm
7	CRP, 0~5 cm
8	CRP, 5~10 cm

Table 1 – Erodible soil sources

Due to the random nature of the physical processes that cause upland erosion, a stochastic un-mixing model is preferred instead of a deterministic one. The objective of this work is to further generalize the Bayesian hierarchical model that was specified by Fox et al. (2004) within the WinBUGS computer software. WinBUGS is a general purpose program for fitting Bayesian models. It was developed by David Spiegelhalter, Andrew, Thomas, Nicky Best, and Wally Gilks at the Medical Research Council Biostatistics Unit, Institute of Public He alth, in Cambridge, UK.

2. MODEL DATA GENERATION

The real source data was available from a previous study for two sources (Corn and Forest) and three tracers ($\delta^{15}N_{AIR}$, $\delta^{13}C_{PDB}$, C/N) for another location. That is, according to the naming convention described above, the data was available for the sources 1, 2, 3 and 4 for three tracers. This data is introduced into this project in order to bring some reality to the data. The data will not be provided due to requirements by the funding organization for that research (NOAA). Data for the fourth tracer for the first four sources was generated using a normal distribution with mean shown in the Table and variance 1. For each of the remaining four sources (5 to 8), 20 samples were generated for all the tracers using a multivariate normal distribution with mean vector shown in the Table 2 and a unit covariance matrix (4 x 4 matrix).

Since no mixture data is available at this moment, mixture data will also be generated by the author. For generating the mixture data, the contribution proportions will be assumed for each source (See Table 3) and these proportions will be multiplied with the mean source vectors to obtain the mean mixture matrix (See Table 4). This will provide a good opportunity to check the performance of the un-mixing model since the real contributions of each source will be known.

Source	$\delta^{15}N_{AIR}$	$\delta^{13}C_{PDB}$	C/N	$\delta^{18}O_{Nitrate}$
1	3.79	-26.42	13.87	23.00
2	3.74	-26.26	13.35	15.00
3	1.39	-25.99	20.30	50.00
4	2.13	-25.42	19.36	30.00
5	5.00	-25.00	13.00	18.00
6	7.00	-27.00	12.00	10.00
7	2.00	-26.00	15.50	40.00
8	2.50	-26.00	15.00	20.00

Table 2 – Mean matrix for sources

Source	Contribution Proportion
1	25
2	8
3	10
4	3
5	25
6	8
7	16
8	5

Table 3 – Assumed contribution proportions of sources

$\delta^{15}N_{AIR}$	$\delta^{13}C_{PDB}$	C/N	$\delta^{18}O_{Nitrate}$
3.7046	-25.9374	14.5863	25.550

Table 4 – Mean matrix for mixture

Ten samples of mixture data is generated using the mean vector in Table 4 and a unit covariance matrix. Data generation for sources and mixture is carried out in Matlab. (See the Appendix)

3. MODEL SPECIFICATION

In order to run the unmixing model in WinBUGS, the paper by Brewer et al. (2003) and final project report by Fox et al. (2004) were used as starting points. Although Brewer's model was used to study water quality rather than sediment deposits, their model is very similar to the one used in this project. Fox et al. (2004) simplified the Brewer's model by changing the log proportions of the source ratios in the model to Dirichlet proportions and applied their model to the sediment deposits. In this project we used Fox's model and modified it so that the model will run with 8 sources and 4 tracers.

Let $\mathbf{X}_{\mathbf{ij}}$ represent the j^{th} observed measured sample data from the i^{th} source (i = 1 to 8). Since the observed data are measurements taken from the sources, it is reasonable to assume that the measurements are normally distributed. Thus, each $\mathbf{X}_{\mathbf{ij}}$ comes from a multivariate normal distribution with mean $\boldsymbol{\mu}_{\mathbf{x,i}}$ and variance-covariance matrix $\boldsymbol{\Sigma}_{\mathbf{x,i}}$. Since, we have 8 sources and 4 tracers, the size of the mean matrix, $\boldsymbol{\mu}_{\mathbf{x,i}}$, is 8x4 and the size of the variance-covariance matrix, $\boldsymbol{\Sigma}_{\mathbf{x,i}}$ is 4x4x8. A vague prior on the variance-covariance matrix is assumed, and conjugate priors are used for efficiency. Therefore, $\boldsymbol{\mu}_{\mathbf{x,i}}$ has a multivariate normal distribution with mean $\boldsymbol{\mu}_{\mathbf{u\ i}}$ and precision matrix $\boldsymbol{\Sigma}_{\mathbf{u,i}}^{\mathbf{-1}}$. These parameters were initialized in the WinBUGS code. $\boldsymbol{\Sigma}_{\mathbf{x,i}}$ was assumed to come from a Wishart distribution with parameters $\boldsymbol{\Omega}_{\mathbf{x,i}}$ and degrees of freedom υ_o.

In the physical problem, the quantity of sediment that truly washed into the stream from the various sources is unobservable. A Bayesian model can still incorporate and simulate the information that can be obtained about this unobservable data. Let $\tilde{\boldsymbol{Y}}_{\mathbf{ij}}$ represent the *j*th unobserved average amount of sediment washed into the stream from the *i*th source. Because this data *could* be measured with the right resources, it can again be assumed that $\tilde{\boldsymbol{Y}}_{\mathbf{ij}}$ is from a multivariate normal distribution with mean $\boldsymbol{\mu}_{\mathbf{x,i}}$ (as defined in the previous paragraph) and variance-covariance matrix $\boldsymbol{\Sigma}_{\tilde{Y},\mathbf{i}}$. This variance-covariance matrix is a derived logical quantity. In the one-dimensional case, a sample average from a population of mean μ and standard deviation σ has a Normal distribution with mean μ and standard deviation $\sigma / (n^{1/2})$, for large *n*. Likewise, since we are working with a vector of averages, the elements of the $\Sigma_{x,i}$ must be multiplied by a factor of n_i, where n_i is assumed to be the number of samples taken from source i. For the present model, n_i is assumed the same as the number of source-soil data values. The matrix $\Sigma_{x,i}$ is the same matrix as defined in the previous paragraph. At the end of the stream, where the mixture takes place between the various sources, we can measure the various amounts of sediments that have worked its way downstream. Let Z_k represent the kth generated data (where k = 1 ,…, 10). Each value of Z_k is a vector of size 4. Since these values are measurements and are subject to measurement error (like $\tilde{Y}_{ij}$ and X_{ij} above), it can again be assumed that the values of Z_k are from a multivariate normal distribution with mean μ_z and variance-covariance matrix Σ_z. Since, we have 4 tracers, the mean matrix, $\boldsymbol{\mu}_{\mathbf{z},}$ is a one dimensional vector of size 4 and the size of the variance-covariance matrix, $\boldsymbol{\Sigma}_{\mathbf{x,i}}$ is 4 x 4. As with the X_{ij} values, conjugate priors were used for efficiency of code. Σ_z comes from a Wishart distribution with parameters $\boldsymbol{\Omega}_z$ and degrees of freedom υ_1. A vague prior was used for this Wishart distribution; $\boldsymbol{\Omega}_z$ was defined as the 4 x 4 identity matrix.

To determine the mean μ_z, several other quantities are needed. We need to know how much sediment was washed downstream (represented by $\tilde{Y}_{ij}$) and the proportion of sediment that came from each source. If P as a vector of proportions where p_i represents the proportion of sediment that came from source I, then the sum $p_1 + p_2 + p_3$ equals 1. As explained in the previous section, multiplying P by $\tilde{Y}_{ij}$, a weighted average of the $\tilde{Y}_{ij}$ values is derived, and can represent the mean μ_z introduced above. Because of the properties of P (specifically that its elements must sum to one), the best choice for deriving these values is from a Dirichlet distribution. These values of P are of primary concern, so we must be able to derive and record these values in the WinBUGS sampler so that appropriate inferences can be made. WinBUGS doesn't let the user directly estimate the parameters of a Wishart prior. In order to overcome this handicap we used a trick for drawing from and recording values from a Dirichlet distribution. This was accomplished using a transformation of random Gamma variables. See Fox et al. (2005) for the transformation. P is a defined logical quantity, which is created from the three Gamma random variables. These Gamma variables can be represented as a vector, which we call δ. Each element in δ is from a Gamma distribution with parameters α_i (a different shape parameter for each indicator) and 1.

4. RESULTS

Three parallel chains of 50,000 iterations are run and the parameters monitored are $\mu_{x,i}$, $\tilde{Y}_{ij}$, P and μ_z. In order to evaluate the sampler performance, we have inspected the history plots, the autocorrelation plots, the Brooks, Gelman and Rubin (BGR) diagnostic and the Monte Carlo (MC) errors.

The history plots for the proportions are shown in Figure 3 with the three chains, represented with different colors. It can be seen that all the three chains oscillate in the same range of values for the P, regardless of the initial values. This random "white noise" pattern gives evidence of convergence of the model. In order to assess the convergence of the model, BGR diagnostic was used for all the chains together (Figure 5). For convergence, the value of BGR ratio should be around 1. Additionally, we have checked that both the pooled and within interval widths converged to stability. In order to evaluate the sampler performance, the autocorrelation plot (Figure 4) was examined. Ideally, all the parameters should drop to zero rapidly. Autocorrelations dropped quickly for P[1] and P[8] but the autocorrelations for other proportions didn't drop rapidly. Fox's model with three tracers and three sources gave high autocorrelations for all the proportions. They concluded that high autocorrelations may be due to large uncertainties that exist in the model and the propagation of these uncertainties in the estimation of the P. However, the MC error (see Table 5 for computed statistics) is very small, supporting the fact that the estimation of the posterior mean of the P is accurate. In fact, as a rule of thumb, the MC error should be smaller than one twentieth of the standard deviation. This condition is largely satisfied for all of the P. It is also important to state that although the MC errors are small and proportion posterior means seem to have converged, posterior standard deviations are relatively high and the uncertainty remains. This is most probably due to the complexity of the model and the proportions' difficulty to be estimated. By its nature, P values can not be observed so no prior data is available and also they can not be estimated directly within the model structure. Figure 6 shows plots of smoothed kernel density for the P. Another important observation would be the model's sensitivity to the shape parameters α of the Gamma distribution from which the Dirichlet distribution is obtained. Results in Table 5 is obtained by using a shape parameter vector of α = c (2.25, 1, 0.8, 1, 2.25, 0.8, 1.4, 0.5) which is quite close to the contribution proportions defined by the user. By comparing mean P values from Table 5 with the real P values from Table 3, one can conclude that the un-mixing model is quite effective if the proportions can be estimated roughly at the beginning and the shape parameters are assigned accordingly.

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
P[1]	0.1956	0.1099	9.003E-4	0.03416	0.1787	0.4529	501	148500
P[2]	0.05175	0.04558	4.017E-4	0.001523	0.03932	0.1688	501	148500
P[3]	0.09334	0.04664	8.358E-4	0.005754	0.09573	0.1771	501	148500
P[4]	0.06527	0.04819	5.784E-4	0.002566	0.05605	0.1778	501	148500
P[5]	0.3338	0.1207	0.001585	0.09927	0.335	0.5621	501	148500
P[6]	0.1279	0.08745	0.001049	0.003549	0.117	0.3197	501	148500
P[7]	0.1091	0.05667	9.386E-4	0.0139	0.1059	0.2233	501	148500
P[8]	0.02323	0.02991	1.895E-4	2.578E-5	0.01158	0.1082	501	148500

Table 5 – Summary statistics for proportions (α = c (2.25, 1, 0.8, 1, 2.25, 0.8, 1.4, 0.5))

In order to check how effective the model is when there is no prior information available about the proportions. The code was run with a shape parameter vector of α = c (1.25, 1.25, 1.25, 1.25, 1.25, 1.25, 1.25, 1.25) which doesn't reflect the proportions that were used to generate the data. P statistics for this case is provided in Table 6. As can be seen, posterior means of the P values are in the right direction and some of the estimates are quite good.

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
P[1]	0.1127	0.08942	6.92E-4	0.00617	0.09078	0.3374	501	148500
P[2]	0.0692	0.05364	5.476E-4	0.003933	0.05651	0.2038	501	148500
P[3]	0.1033	0.04636	7.924E-4	0.01461	0.1045	0.1884	501	148500
P[4]	0.06696	0.04698	5.244E-4	0.004365	0.05822	0.1772	501	148500
P[5]	0.2584	0.1305	0.001979	0.03074	0.2546	0.5151	501	148500
P[6]	0.2129	0.1	0.001389	0.0329	0.2103	0.4122	501	148500
P[7]	0.1166	0.05828	9.53E-4	0.01477	0.1147	0.2331	501	148500
P[8]	0.0599	0.04648	2.956E-4	0.003342	0.04912	0.1755	501	148500

Table 6 – Summary stats for proportions (α=c (1.25, 1.25, 1.25, 1.25, 1.25, 1.25, 1.25, 1.25))

Plots in the Appendix are for the case with α = c (2.25, 1, 0.8, 1, 2.25, 0.8, 1.4, 0.5) and the plots for second case which are not provided in this paper also showed that the convergence criterions were satisfied for that specific case.

5. FIGURES

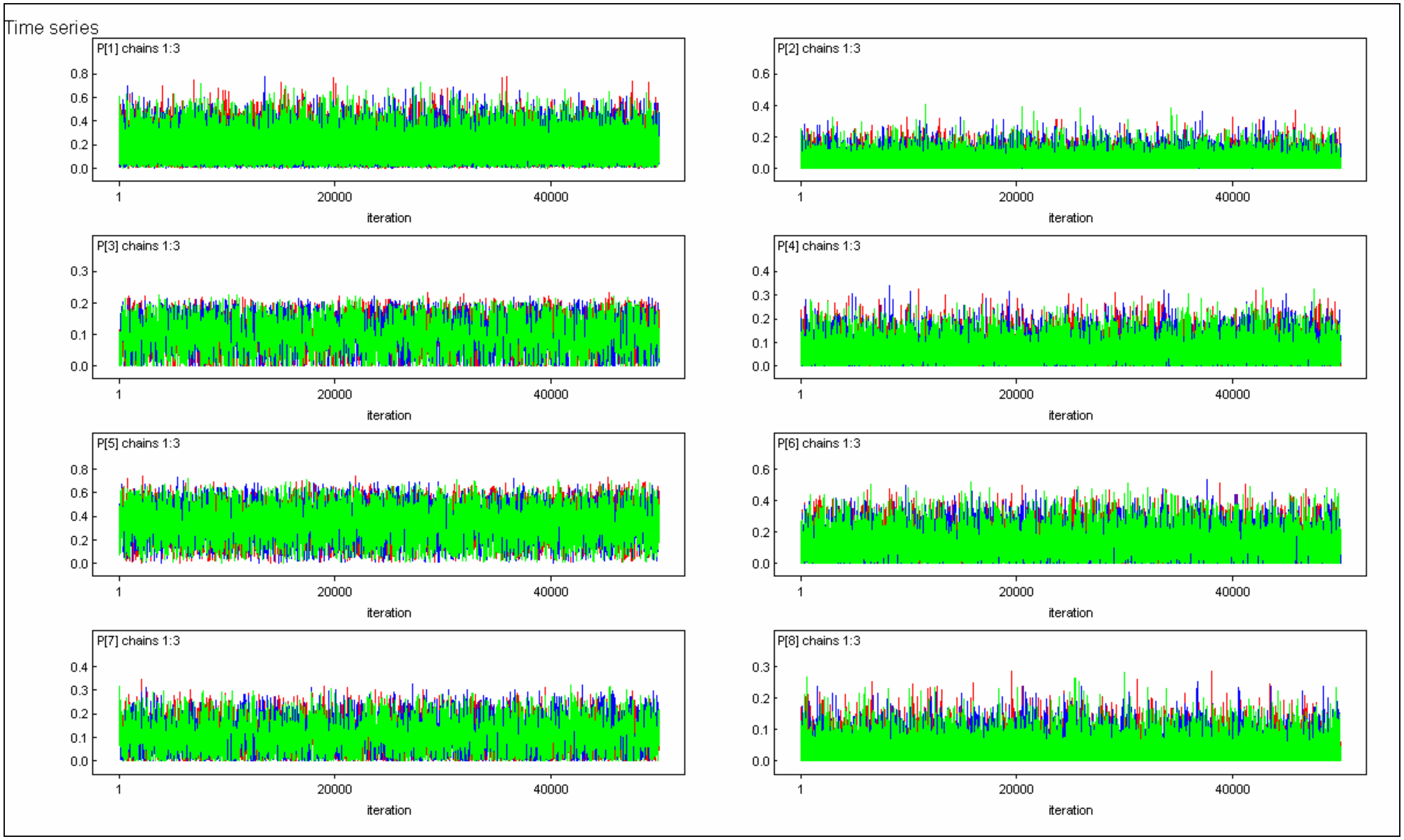

Figure 3 - History plots for the three chains for P for all the eight proportions

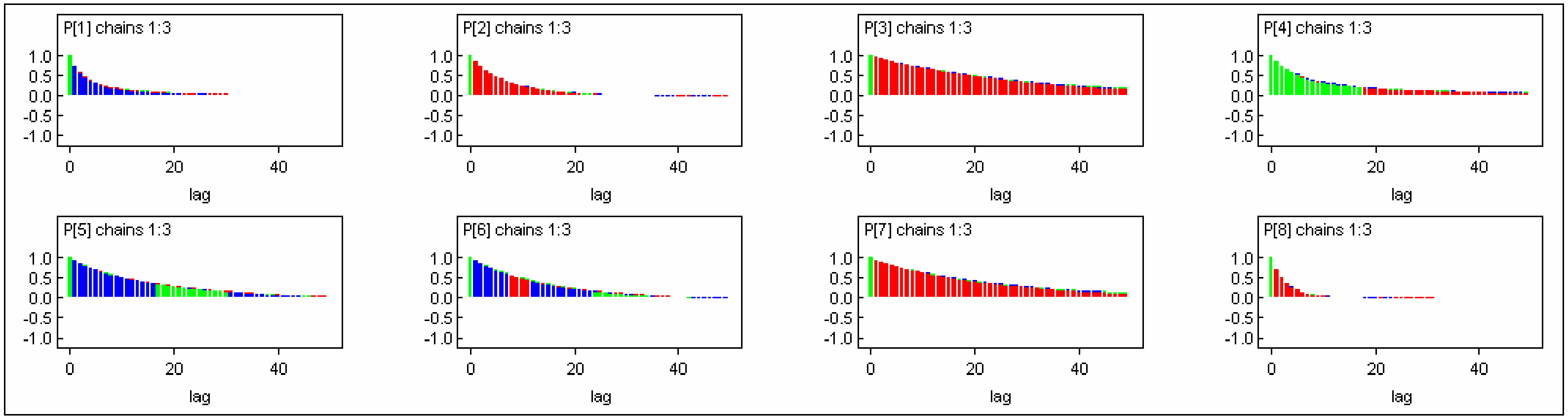

Figure 4 - Autocorrelation plots for P

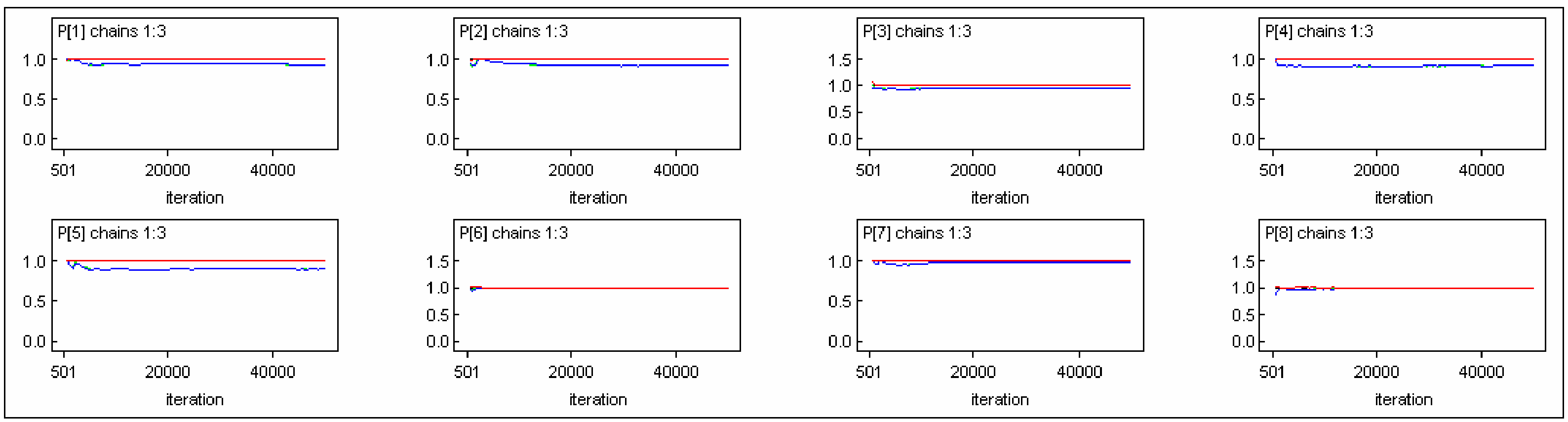

Figure 5 - BGR diagnostic plots

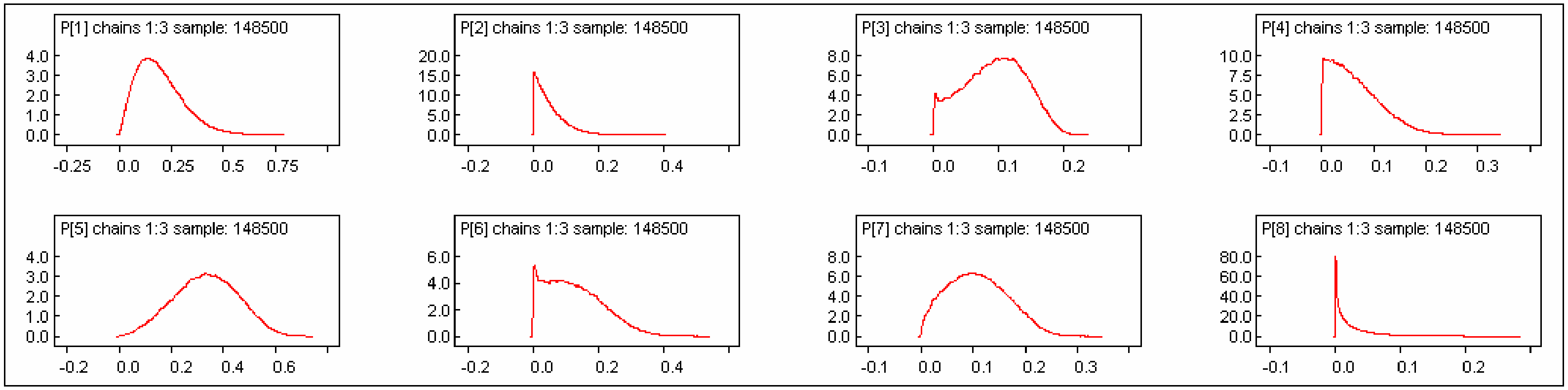

Figure 6 - Plots of the smoothed kernel density for the P

6. APPENDIX

6.1. WinBUGS code

```
# i = sources
# j = samples within the sources
# k= mixture samples
# M= number of sources
# N[i]= number of samples per source
# T= number of mixture samples

model
{
	for( r in 1 : sumN ) {
		x[r, 1:4] ~ dmnorm(mu[source[r], ], epsX[source[r], ,])
	}
	for( i in 1 : M ) {
		mu[i, 1:4] ~ dmnorm(muu[i , ],precmu[i, ,])
		epsX[i,1:4, 1:4] ~ dwish(Omega[i, , ], 9)
		ybar[i, 1:4] ~ dmnorm(mu[ i , ], epsYbar[i, ,])
		for(w in 1:4){
			for(f in 1:4){
				epsYbar[i, w, f] <- epsX[i, w, f]*N[i]
}
		}
	}
	for( k in 1 : T ) {
		z[k, 1:4] ~ dmnorm(muz[], epsZ[ , ])
}
for(q in 1:4) {
muz[q] <- inprod(P[],ybar[ , q])
}
	for (k in 1:8) {
	P[k] <- delta[k] / sum(delta[])
	delta[k] ~ dgamma(alpha[k], 1)
}
	epsZ[1:4, 1:4] ~ dwish(OmegaZ[ , ], 9)
}

#data
list(M = 8,
N = c(87,54 ,71, 33, 20, 20, 20, 20),
sumN=325, T = 10,
muu = structure(.Data = c(3, -25, 13, 23,
3, -25, 13, 14,
1, -25, 21, 52,
3, -25, 18, 32,
5, -26, 13, 19,
```

```
6, -28, 11, 11,
3, -29, 14, 40,
1, -27, 16, 22),
.Dim = c(8, 4)),
precmu = structure(.Data = c(1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1,
1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1,
1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1,
1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1,
1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1,
1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1,
1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1,
1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1),
.Dim = c(8, 4, 4)),
Omega = structure(.Data = c(1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1,
1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1,
1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
```

```
0, 0, 0, 1,
1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1,
1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1,
1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1,
1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1,
1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1),
.Dim = c(8, 4, 4)),
alpha = c(2.25, 1, 0.8, 1, 2.25, 0.8, 1.4, 0.5),
OmegaZ = structure(.Data = c(1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1),
.Dim = c(4, 4)),
z = structure(.Data = c(4.8358, -27.407, 14.31, 25.828,
2.7803, -26.473, 14.546, 26.997,
5.1221, -25.706, 14.425, 24.966,
3.279, -23.452, 15.165, 24.741,
4.6029, -27.497, 13.699, 25.391,
4.7133, -23.862, 14.847, 24.847,
4.128, -25.959, 13.705, 25.653,
4.4349, -25.045, 14.184, 24.661,
3.6133, -25.802, 14.44, 23.24,
4.4714, -24.251, 14.991, 24.578),
.Dim = c(10, 4)))

x[,1] x[,2] x[,3] x[,4] source[]
4.00    -26.21  13.42   24.72   1
4.56    -26.49  13.63   22.73   1
.
.
.
3.22    -27.18  15.07   19.57   8
2.73    -25.68  15.19   18.52   8
```

```
3.21	-26.81	14.70	20.92	8
3.22	-25.54	15.28	18.10	8
2.14	-26.26	14.11	20.28	8
3.74	-24.97	13.75	18.45	8
2.13	-26.04	16.13	20.89	8
2.43	-25.48	14.52	19.88	8
END

#INITIAL VALUES (Chain 1)
list(
mu = structure(.Data = c(3.79, -26.42, 13.87, 23,
3.74, -26.26, 13.35, 15,
1.39, -25.99, 20.30, 50,
2.13, -25.42, 19.36, 30,
5, -25, 13, 18,
7, -27, 12, 10,
2, -26, 15.5, 40,
2.5, -26, 15, 20),
.Dim = c(8, 4)),
delta = c(1, 1, 1, 1, 1.5, 1.5, 1.5, 1.5),
epsX = structure(.Data = c(1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1,
1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1,
1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1,
1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1,
1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1,
1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1,
1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1,
```

```
1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1),
.Dim = c(8, 4, 4)),
epsZ = structure(.Data = c(1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1),
.Dim = c(4, 4)))

#INITIAL VALUES (Chain 2)
list(
mu = structure(.Data = c(3.79, -26.42, 13.87, 23,
3.74, -26.26, 13.35, 15,
1.39, -25.99, 20.30, 50,
2.13, -25.42, 19.36, 30,
5, -25, 13, 18,
7, -27, 12, 10,
2, -26, 15.5, 40,
2.5, -26, 15, 20),
.Dim = c(8, 4)),
delta = c(2.25, 1, 0.8, 1, 2.25, 0.8, 1.4, 0.5),
epsX = structure(.Data = c(1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1,
1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1,
1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1,
1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1,
1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1,
1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1,
1, 0, 0, 0,
```

```
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1,
1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1),
.Dim = c(8, 4, 4)),
epsZ = structure(.Data = c(1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1),
.Dim = c(4, 4)))

#INITIAL VALUES (Chain 3)
list(
mu = structure(.Data = c(3.79, -26.42, 13.87, 23,
3.74, -26.26, 13.35, 15,
1.39, -25.99, 20.30, 50,
2.13, -25.42, 19.36, 30,
5, -25, 13, 18,
7, -27, 12, 10,
2, -26, 15.5, 40,
2.5, -26, 15, 20),
.Dim = c(8, 4)),
delta = c(1.25, 1.25, 1.25, 1.25, 1.25, 1.25, 1.25, 1.25),
epsX = structure(.Data = c(1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1,
1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1,
1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1,
1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1,
1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1,
1, 0, 0, 0,
0, 1, 0, 0,
```

```
0, 0, 1, 0,
0, 0, 0, 1,
1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1,
1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1),
.Dim = c(8, 4, 4)),
epsZ = structure(.Data = c(1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1),
.Dim = c(4, 4)))
```

6.2. MATLAB Code

Data generation from a MVN(mu,sigma) for each of the sources 5 to 8 (Code given only for source 5):

```
mu = [7 -27 12 10];
sigma = [1 0 0 0; 0 1 0 0; 0 0 1 0; 0 0 0 1];
r = mvnrnd(mu,sigma,20);
```

Computation of mean mixture vector:

Mean source matrix, x is

```
x = [3.7900    3.7400    1.3900    2.1300    5.0000    7.0000    2.0000    2.5000
-26.4200  -26.2600  -25.9900  -25.4200  -25.0000  -27.0000  -26.0000  -26.0000
13.8700   13.3500   20.3000   19.3600   13.0000   12.0000   15.5000   15.0000
23.0000   15.0000   50.0000   30.0000   18.0000   10.0000   40.0000   20.0000];
```

Proportions, y is

```
y = [0.25
0.08
0.10
0.03
0.25
0.08
0.16
```

0.05];

m = x*y

Output is,

m =
3.7046
-25.9374
14.5863
25.5500

Mixture data creation:

```
mu=[3.7046    -25.9374    14.5863    25.55];

sigma=[1    0    0    0
0    1    0    0
0    0    1    0
0    0    0    1];

mixture=mvnrnd(mu,sigma,10);
```