Cah Techn. Inra, 2010, 69, 41-49

MMIX : un package R pour combiner des modèles en agronomie

Marie Morfin et David Makowski[1]

Résumé : *MMIX est un package qui peut être utilisé avec le logiciel libre R pour évaluer, sélectionner et combiner des modèles linéaires et logistiques. Cet article présente brièvement les méthodes mises en œuvre par ce package et montre comment il peut aider les agronomes à analyser leurs données expérimentales. L'utilisation du package est illustrée dans une application dont l'objectif est l'identification des facteurs limitants du nombre de grains du blé biologique.*

Mots clés : bootstrap, combinaison de modèles, modèle linéaire généralisé, statistique bayésienne

Introduction

Les modèles agronomiques fournissent des informations utiles aux agriculteurs, instituts techniques, et décideurs politiques. Pour un problème pratique donné, plusieurs modèles plus ou moins complexes sont généralement disponibles pour prédire une variable d'intérêt en fonction des pratiques agricoles et des caractéristiques du milieu comme, par exemple, le rendement d'une culture ou le risque de pollution de l'eau par les nitrates. Dans ce genre de situation, une approche classique consiste à identifier un modèle unique à l'aide d'une méthode statistique de sélection. Toutes les applications sont ensuite réalisées à partir du modèle sélectionné. Diverses méthodes de sélection sont habituellement utilisées par les scientifiques (e.g. validation croisée, stepwise) mais, en général, l'incertitude de la procédure de sélection est ignorée une fois le modèle sélectionné. Certains statisticiens ont pourtant montré que, lorsque les erreurs des modèles sont grandes, une légère modification de la base de données peut conduire à des résultats de sélection complètement différents des résultats initiaux. Des statisticiens ont également montré qu'il pouvait être préférable de combiner tous les modèles existant plutôt que de n'utiliser qu'un seul modèle.

Dans le cadre des régressions linéaires simples ou logistiques, plusieurs méthodes de combinaison de modèles ont été développées par les statisticiens : Bayesian Model Averaging (Raftery *et al.,* 1997, Viallefont *et al.,* 2001), AIC-based mixing (Burnham K.P. and Anderson DR, 2002), Adaptative Regression by Mixing with a model Screening step (Yuan Z. and Yang Y., 2005 et Yuan Z. and Ghosh D., 2008).

Le package MMIX, créé pour le logiciel statistique R, permet aux agronomes de mettre facilement en œuvre ces méthodes, ainsi que des outils pour les comparer aux méthodes classiques de sélection, comme la sélection stepwise. Le package MMIX permet également d'évaluer la stabilité des estimations par re-échantillonage bootstrap ainsi que la qualité des prédictions des modèles par validation croisée. Cet article présente le principe général de ces méthodes et le mode d'emploi du package MMIX à l'aide d'un cas d'étude sur du blé biologique.

[1] UMR211- INRA AgroParisTech - F-78850 Thiverval-Grignon
☏ 33 (0)1 30 81 59 92 ✉ david.makowski@grignon.inra.fr

1. Méthodes de combinaison de modèles

Dans le cadre d'une régression linéaire, l'espérance de la variable d'intérêt Y est reliée à un ensemble de p variables explicatives X = (X_1, . . . ,X_p) par :

$$\mathbf{E(Y \mid X) = \theta_0 + \theta_1 X_1 + \cdots + \theta_i X_i + \cdots + \theta_p X_p}$$

où $\theta_0, \ldots, \theta_1$ sont les paramètres du modèle.
Dans le cadre d'une régression logistique, la relation entre Y et les p variables explicatives est définie par :

$$\mathbf{logit\big(E(Y \mid X)\big) = \theta_0 + \theta_1 X_1 + \cdots + \theta_i X_i + \cdots + \theta_p X_p}$$

Les modèles définis ci-dessus correspondent aux modèles complets, i.e. les modèles incluant le jeu complet des variables explicatives X_1, . . . ,X_p. À partir de ces variables on peut définir d'autres modèles, reliant la variable d'intérêt Y à un sous-ensemble des variables explicatives. Les modèles définis à partir de tous les sous-ensembles possibles de X seront notés par la suite M_1, ..., M_K, où $K=2^p$.
Les trois méthodes de combinaisons de modèles mises en œuvre par MMIX permettent de calculer des poids w_k pour chacun des K modèles possibles, puis d'estimer les paramètres par :

$$\hat{\theta}_i = \sum_{k=1}^{K} w_k \hat{\theta}_i^{(k)}$$

où $\hat{\theta}_i^{(k)}$ est l'estimateur par maximum de vraisemblance du paramètre $\hat{\theta}_i$ dans le modèle M_k.
Les méthodes diffèrent sur la technique utilisée pour calculer les poids. La somme des poids de l'ensemble des modèles incluant une variable X donnée correspond à la probabilité que X ait une influence sur la réponse Y (égale elle-même à la probabilité que le paramètre associé à X soit non nul). Ces probabilités permettent de juger de l'importance de l'effet des variables explicatives X sur Y.

1.1 Bayesian Model Averaging

La méthode de Bayesian Model Averaging a été présentée par Raftery *et al.*, (1997), Hoeting *et al.*, (1999) et, Viallefont *et al.,* (2001).
La fonction bmaBic() de MMIX implémente cette méthode en utilisant le BIC (Bayesian Information Criterion). Le poids du modèle Mk est ainsi calculé par :

$$w_k^{BIC} = \frac{\exp\left(-BIC_k / 2\right)}{\sum_{l=1}^{K} \exp\left(-BIC_l / 2\right)}$$

où BIC_k est le BIC du modèle M_k.
La variance de l'estimateur non conditionnée sur le choix du modèle est estimée par :

$$\hat{Var}\left(\hat{\theta}_i\right) = \sum_{k \in \Gamma_i} w_k^{BIC} \left[\hat{\sigma}_i^{(k)2} + \left(\hat{\theta}_i^{(k)} - \hat{\theta}_i\right)^2 \right]$$

où $\hat{\sigma}_i^{(k)}$ est l'estimation de l'écart-type de l'estimateur $\hat{\theta}_i^{(k)}$, et Γ_i est l'ensemble des modèles incluant la $i^{ème}$ variable explicative.

La fonction BMA() (Raftery *et al.* (2005)) est une autre fonction R qui permet de caculer ces poids, mais la fonction bmaBic() de MMIX inclut un nombre réduit d'arguments et elle est plus facile à utiliser.
La sortie de bmaBic() est une liste contenant les estimations des paramètres (coef), les probabilités que chaque paramètre soit non nul (pne0), les écarts-types non conditionnés des paramètres (sd), les prédictions des observations (fitted.values), le poids des modèles (modweights) ainsi que leurs paramètres (allcoef), la liste des modèles (label), et les trois meilleurs modèles vis à vis du critère BIC (BestModels). La fonction summary( ) affiche l'ensemble des résultats. La fonction plot() génère un diagramme en bâton des probabilités pne0.

1.2 AIC-based mixing

Burnham K.P. and Anderson D.R., (2002) présente une technique de combinaison de modèles où les poids des modèles sont calculés à partir du critère AIC (Akaike Information Criterion) :

$$w_k^{AIC} = \frac{\exp\left(-AIC_k / 2\right)}{\sum_{l=1}^{K} \exp\left(-AIC_l / 2\right)}$$

w_k^{AIC} est le poids du modèle M_k, et AIC_k et la valeur du critère AIC pour le modèle M_k. Cette technique est mise en œuvre par MMIX à l'aide de la fonction mixAic(). Cette fonction génère les mêmes types de résultats que bmaBic().

1.3 Adaptative Regression by mixing with a Model Screening step

The Adaptative Regression by mixing with a Model Screening step (ARMS) a été proposé par Yuan et Yang (2005) pour le modèle linéaire simple et par Yuan et Ghosh (2008) pour la régression logistique. Etant donné un échantillon de n observations, les poids des modèles sont calculés à l'aide de l'algorithme suivant :

1. Un jeu de données d'entraînement J_1 de taille [n/2] (= n/2 si n est pair) est tiré aléatoirement à partir du jeu de données complet.
2. Les m meilleurs modèles sont sélectionnés selon les critères AIC et/ou BIC.
3. Les paramètres des m modèles sont estimés à partir de l'échantillon J_1. Pour le modèle linéaire, l'écart-type des erreurs résiduelles est également estimé à cette étape.
4. Les poids w_k des m modèles sont calculés à partir de la seconde partie de l'échantillon J_2, à l'aide des paramètres estimés à l'étape 3.
5. Les quatre étapes précédentes sont répétées N fois, et le poids de chaque modèle M_k est finalement la moyenne de leur poids calculés sur les N itérations:

$$\overline{w}_k = \frac{1}{N} \sum_{l=1}^{N} w_k(l)$$

où $w_k(l)$ est le poids du modèle M_k calculé à la $l^{ème}$ itération.

La probabilité qu'un paramètre θ_i soit non nul se calcule directement à partir des poids des modèles: $\hat{P}(\theta_i \neq 0) = \sum_{M_k \in \Gamma_i} \overline{w}_k$

où Γ_i est l'ensemble des modèles contenant la variable explicative X_i.

La fonction arms() du package MMIX applique cet algorithme avec les arguments suivants :

data : un tableau de données comprenant la variable d'intérêt et les variables explicatives. Toutes ces variables doivent être numériques et la variable d'intérêt prend la valeur 0 ou 1 pour le modèle logistique ;

family : description de l'erreur de distribution (gaussienne ou binomiale) ;

nsample : nombre d'itérations dans l'algorithme (N) ;

nbest : nombre de modèles sélectionnés à l'étape de screening (m) ;

criterion : critère de sélection utilisé à l'étape de screening: "aic", "bic" or "both". "both" signifie que nbest modèles sont sélectionnés par chacun des critères ;

weight : type de poids de modèle, "likeli" pour les likelihood-weights ou "aic" pour les AIC-weights.

maxVar: nombre maximum de variables explicatives à inclure dans le modèle. Si maxVar < p, maxVar variables seront sélectionnés au maximum dans une étape préliminaire à l'algorithme par une sélection stepwise dans la direction « forward », avec le critère AIC.

La fonction arms() génère un tableau comprenant les estimations des paramètres par ARMS (coef) et leur probabilité d'être non nul (pne0).

La fonction plot() trace un diagramme des probabilités pne0 associées aux variables explicatives, et des résultats complémentaires tels que les prédictions du jeu de données, sont donnés par la fonction summary().

La sortie est également une liste à partir de laquelle coef, pne0, les prédictions (fitted.values), les variables explicatives de chaque modèle (label), le poids de chaque modèle (modweights), et une matrice des estimations des paramètres de tous les modèles (allcoef) sont accessibles.

2. Fonction bootFreq()

La fonction bootFreq() évalue la stabilité des sélections stepwise et des méthodes de combinaison de modèles par bootstrap (Prost L., Makowski D. and Jeuffroy M.-H., 2008). Cette fonction génère des échantillons à partir du jeu de donnée initial en tirant aléatoirement des jeux de données de la même taille avec remise (Efron B. and Tibshirani R.J., 1993).

Les méthodes de sélection stepwise et de combinaison de modèles sont alors appliquées à chacun des échantillons bootstrap afin de calculer la fréquence de sélection de chaque variable explicative et la déviation standard des estimateurs des coefficients sur l'ensemble des échantillons. Une fréquence de sélection proche de zéro ou un signifie que les résultats sont stables tandis qu'une fréquence proche de 0,5 traduit une instabilité de la sélection des variables pour ce jeu de données.

La fonction bootFreq() met en oeuvre cette approche avec les arguments suivants :

data : un tableau de données comprenant la variable réponse et les variables explicatives. Toutes les variables doivent être numériques et la variable réponse doit valoir 0 ou 1 pour le modèle logistique ;

family : une description de la distribution de l'erreur (gaussienne ou binomiale) ;

nboot : le nombre d'échantillons bootstrap tirés ;

method : la méthode statistique utilisée pour estimer les paramètres du modèle. method=1 pour fullModel (les paramètres du modèle sont estimés par maximum de vraisemblance sans aucune sélection de variables, method=2 pour stepSel (les variables sont sélectionnées par une sélection stepwise), method=3 pour bmaBic, method=4 pour mixAic, method=5 pour arms ;

file : le chemin du fichier où se trouve les résultats qui seront stockés au fur et à mesure que la fonction tourne. Si file = NULL aucun fichier ne sera créé ;

Par ailleurs, les arguments spécifiques à la méthode demandée doivent être ajoutés dans l'instruction bootFreq();

bootFreq( ) retourne un objet de classe "classMMIX". Un tableau de données où figurent les résultats principaux est affiché et la fonction plot() permet d'obtenir un diagramme des fréquences de sélection de chaque variable explicative. Une sortie bootFreq() est également une liste contenant un vecteur des fréquences de sélection de chaque variable (frequency), les valeurs des estimations des paramètres pour tous les échantillons bootstrap (coef), la valeur moyenne des estimations des paramètres sur tous les échantillons bootstrap, l'écart-type des estimations des paramètres sur tous les échantillons bootstrap (sd) et pne0, les moyennes des probabilités des variables pour les méthodes de combinaison de modèles, équivalents aux fréquences de sélection pour les autres méthodes.

3. Evaluation des modèles

MMIX propose également des méthodes pour comparer la performance de modèles sélectionnés par sélection stepwise et des combinaisons de modèles. Le critère PMSE (Predictive Mean Square Error) est utilisé pour le modèle linéaire tandis que le critère AUC (Area Under Curve) est utilisé pour le modèle logistique.
La fonction pmseCV() estime le PMSE par validation croisée « leave-np-out », c'est-à-dire que l'on enlève np observations du jeu de données initial pour l'estimer les paramètres puis on calcule l'erreur de prédiction sur ces np données.
Pour les modèles logistiques, la fonction aucCV() estime l'AUC par une validation croisée "leave-np-pair-out". Les observations sont enlevées par paires d'individus, un de chaque modalité de la variable réponse (0 et 1). L'AUC est estimé à chaque itération par la statistique de Wilcoxon.
Pour ces deux fonctions, les arguments sont :

data : un tableau de données comprenant la variable réponse et les variables explicatives. Toutes les variables doivent être numériques et la variable réponse doit valoir 0 ou 1 pour le modèle logistique.

family : une description de la distribution de l'erreur (gaussienne ou binomiale).

method : la méthode statistique utilisée pour estimer les paramètres du modèle. method=1 pour fullModel (les paramètres du modèle sont estimés par maximum de vraisemblance sans aucune sélection de variables), method=2 pour stepSel (les variables sont sélectionnées par une sélection stepwise), method=3 pour bmaBic, method=4 pour mixAic, method=5 pour arms.

np : le nombre d'individus (pmseCV) ou paires d'individus (aucCV) retirés à chaque étape de la validation croisée.

random : les observations sont sélectionnées aléatoirement si random=TRUE. random peut être FALSE seulement si np=1. Dans ce cas tous les individus seront retirés au cours de la validation croisée.

Npermu : si random=TRUE, le nombre d'échantillons de np individus tirés aléatoirement.

file : le chemin du fichier où se trouvent les résultats qui seront stockent au fur et à mesure que la fonction tourne. A chaque itération les np prédictions (première colonne) et les np observations correspondantes (deuxième colonne) y sont enregistrées. Si file = NULL aucun fichier ne sera créé.

Les arguments spécifiques à la méthode demandée doivent être également ajoutés.

4. Exemple : nombre de grains de blé

Nous présentons ici les résultats d'une application du package MMIX réalisée pour illustrer les méthodes de combinaison de modèles linéaires. Des expérimentations ont été effectuées en France sur 16 sites-années de blé d'hiver biologique (Casagrande M. *et al.,* 2009). L'objectif est de modéliser le nombre de grains par m^2 (GN) en fonction de neuf variables agronomiques mesurées sur chaque site-année : le bilan hydrique pendant la période de croissance végétative (WBV), le quotient photothermique pendant la période de croissance végétative (PQV), et après floraison (PQG), la densité de mauvaises herbes (WD), l'index de nutrition azotée (NNI), le type de variété (BAF), la compression du sol (SC).
Des modèles de régression linéaire ont été ajustés aux données afin d'identifier les variables qui expliquent la variable réponse GN. Les paramètres du modèle ont été estimés par plusieurs méthodes : le modèle linéaire incluant toutes les variables explicatives (modèle complet), la sélection stepwise dans la direction "both" et selon les critères AIC (stepwise Aic) et BIC (stepwise Bic), bayesian model averaging (bmaBic), AIC-based mixing (mixAic), et ARMS en utilisant les deux critères de sélection AIC et BIC, et les poids de type likelihood (armsL) et AIC (armsA). Les résultats sont résumés dans la **figure 1** et le **tableau 1**. Les instructions R utilisées pour appliquer MMIX sont présentées en annexe.
Seulement deux variables (NNI et BAF) ont été sélectionnées par les méthodes de sélection stepwise. Les probabilités associées à ces deux variables varient entre 0,5 et 1 selon la méthode utilisée. La variable WD obtient également un poids supérieur à 0,5 avec la méthode mixAic (**figure 1**). Les performances des différentes méthodes ont été comparées en calculant leur PMSE avec la fonction pmseCV. Les meilleures prédictions sont obtenues avec bmaBic, suivie de près par mixAic et armsL. Le modèle complet et les sélections stepwise prédisent moins bien pour ce jeu de données.

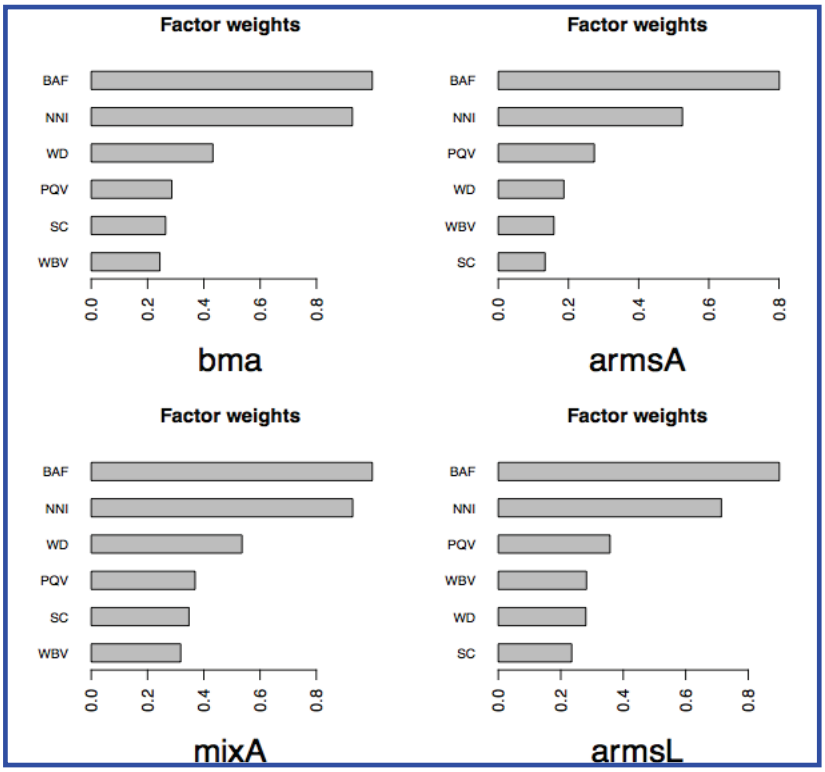

Figure 1: Probabilités que les variables explicatives aient un effet sur le nombre de grains du blé. Probabilités calculées avec 4 méthodes de combinaison de modèles mises en œuvre par MMIX (bmaBic, mixAic, armsL et armsA) pour le jeu de données *blé biologique*. Ces figures ont été obtenues avec l'instruction plot de MMIX. Voir annexe.

		Intercept	WBV	PQV	WD	NNI	BAF	SC
Modèle complet	coef	4943	-3.10	1882	-9	14226	-6283	-929
Stepwise Aic	coef	3626	0	0	0	18673	-6239	0
Stepwise Bic	coef	3626	0	0	0	18673	-6239	0
bmaBic	coef	4534	2.6	727	-3.7	16283	-6244	-273
	pne0	1.0	0.2	0.3	0.4	0.9	1.0	0.3
	Sd	4009	41.6	2142.5	6.1	7605	1359	932
mixAic	coef	4597	2.4	931	-4.6	15950	-6255	-364
	pne0	1.0	0.3	0.4	0.5	0.9	1	0.3
	Sd	4071	48	2435	6.5	7766	1370	1074
arms L	coef	6384	-6	1395	-2,00	9369	-4807	-113
	pne0	1.0	0.2	0.3	0.2	0.5	0.8	0.1
arms A	coef	4939	-4	1479	-3	12802	-5482	-189
	pne0	1.0	0.3	0.4	0.3	0.7	0.9	0.2

Tableau 1: Résultats obtenus avec le modèle linéaire complet, deux méthodes stepwises, les fonctions bmaBic et mixAic, et la fonction arms (avec les poids Likelihood et Aic). Les valeurs de "coef" correspondent aux valeurs estimées des paramètres, les valeurs de "pne0" indiquent les probabilités que les paramètres soient différents de zéro, et "sd" correspond à l'écart-type. Voir l'annexe pour le code informatique.

Conclusion

Le package MMIX propose des fonctions compatible avec le logiciel statistique R pour estimer les paramètres de modèles linéaires et logistiques à l'aide de techniques de régressions stepwise et des méthodes de combinaison de modèles. Il comprend également des fonctions pour évaluer la stabilité et comparer les performances de ces différentes méthodes. Le package MMIX et ses notices sont disponibles gratuitement sur les sites web suivants :
http://cran.r-project.org/web/packages/#available-packages-M
http://www.versailles-grignon.inra.fr/agronomie/productions/logiciels/mmix_an_r_package

Remerciements : Les auteurs remercient C. Cadet, M. Casagrande et C. Vallée de leur contribution à ce projet. Ce travail a été financé par l'Agence nationale pour la recherche (ANR-JCJC).

Bibliographie

Burnham K.P. and Anderson D.R. (2002) Model selection and multimodel inference: a practical information-theoric approach. Springer, 2nd edition

Casagrande M., David C., Valantin-Morison M., Makowski D. and Jeuffroy M.-H. (2009) Factors limiting the grain protein content of organic winter wheat in south-eastern France: a mixed-model approach. Agronomy for Sustainable Development 29: 565-574

Efron B. and. Tibshirani R.J. (1993) An introduction to the bootstrap. New-York: Chapman & Hall/CRC.

Hoeting J.A., Madigan D., Raftery A.E. and Volinsky C.T. (1999) Bayesian model averaging: a tutorial. Statistical science, 14:382-417

Prost L., Makowski D. and. Jeuffroy M.-H. (2008) Comparison of stepwise selection and Bayesian model averaging for yield gap analysis. Ecological Modelling, 219:66-76

Raftery A.E., Madigan D. and Hoeting J.A. (1997) Bayesian model averaging for linear regression models. Journal of the American statistical association, 92:179-191

Raftery A.E., Painter I.S. and Volinsky C.T. (2005) BMA: an R package for Bayesian model averaging. R News, 5:2-8

Viallefont V., Raftery A.E. and Richardson S. (2001) Variable selection and Bayesian model averaging in case-control studies. Statistics in medicine, 20:3215-3230

Yuan Z. and Ghosh D. (2008) Combining Multiple Biomarker Models in Logistic Regression. Biometrics, 64:43-439

Yuan Z. and Yang Y. (2005) Combing Linear Regression Models: When and How? Journal of the American statistical association, 100:1202-1214

Annexe

Commandes R utilisées avec les données « blé biologiques »: application de cinq méthodes d'estimation, plot des probabilités que les variables aient un effet, analyse bootstrap pour les méthodes de sélection stepwise, et estimation des PMSE.

```
M1 <- fullModel(data=tabGN, family=gaussian("identity"))
M2 <- stepSel(data=tabGN, family=gaussian("identity"), direction="both", criterion="aic")
M3 <- stepSel(data=tabGN, family=gaussian("identity"), direction="both", criterion="bic")
M4 <- bmaBic(data=tabGN, family=gaussian("identity"))
M5 <- mixAic(data=tabGN, family=gaussian("identity"))
M6 <- arms(data=tabGN, family=gaussian("identity"), nbest=40, criterion="both", weight="aic")
M7 <- arms(data=tabGN, family=gaussian("identity"), nbest=40, criterion="both", weight="likeli")

B2 <- bootFreq(data=tabGN, family=gaussian('identity'), nboot=500, method=2, criterion='aic', trace=0)
B3 <- bootFreq(data=tabGN, family=gaussian('identity'), nboot=500, method=2, criterion='bic',trace=0)

par(mfcol=c(2,2))
plot(M4)
title(sub="bma")
plot(M5)
title(sub="mixaic")
plot(M6)
title(sub="arms aic")
plot(M7)
title(sub="arms likeli")

P1 <- pmseCV(data=tabGN, method=1, np=1, random=F)
P2 <-pmseCV(data=tabGN, method=2,np=1,criterion="aic", random=F, trace=0)
P3 <-pmseCV(data=tabGN, method=2,np=1, criterion="bic", random=F)
P4 <- pmseCV (data=tabGN, method=3, random=F,np=1)
P5 <- pmseCV (data=tabGN, method=4, random=F,np=1)
P6 <- pmseCV (data=tabGN, method=5, weight="aic", nbest=40, nsample=20, random=F, np=1)
P7 <- pmseCV (data=tabGN, method=5, weight="likeli", nbest=40, nsample=20, random=F, np=1)
rpmseCv <-sqrt(rbind(P1, P2, P3, P4, P5, P6, P7)[,1])
```