Le tableau suivant **[81]** compare les trois techniques computationnelles majeures évoquées.

Tableau 4 : Etude comparative des techniques ab initio, semi –empirique et mécanique moléculaire.

Ab initio	**Semi -empirique**	**Mécanique moléculaire**
• Prise en compte de tous les électrons. • Limitée à quelques dizaines d'atomes. • Nécessite un super calculateur. • Peut être appliquée à des composés inorganiques, organique,organométalliques ,et aux fragments moléculaires (composants catalytiques d'enzymes). • Vide, solvatation implicite. • Applicable à l'état fondamental, et aux états de transition et excité.	• Ignore certains électrons (simplification). • Limitée à quelques centaines d'atomes. • Peut être appliquée à des composés inorganiques, organiques, organométalliques et de petits oligomères (peptides, nucléotides, saccharides). • Vide, solvatation implicite. • Applicable à l'état fondamental, et aux états de transition et excité.	• Ignore tous les électrons ,seuls les noyaux sont considérés. • Molécules contenant des milliers d'atomes. • Peut être appliquée aux composés inorganiques, organiques, oligo–nucléotides, peptides, saccharides,métallo-organiques et inorganiques. • Vide, solvatation implicite ou explicite. • Applicable uniquement à l'état fondamental.

IV-1 Modélisation

La modélisation par apprentissage consiste à trouver le jeu de paramètres qui conduit à la meilleure approximation possible de la fonction de régression, à partir des couples entrées/ sortie constituant l'ensemble d'apprentissage (ou de calibrage) ; le plus souvent, ces couples sont constitués d'un ensemble de vecteurs de variables (descripteurs dans le cas de molécules) { x^i, i = 1……..n}, et un ensemble de mesures de la grandeur à modéliser {y (x^i), i= 1…….n} **[53]**. La détermination des valeurs de ces paramètres nécessite la mise en œuvre de méthodes d'optimisation qui diffèrent selon le type de modèle choisi.
Dans cette thèse deux types de méthodes ont été exploitées.

IV-2 Régression linéaire multiple (MLR) [82-84]

La régression linéaire multiple est la méthode la plus simple de modélisation, elle consiste à rechercher une équation linéaire par rapport à ses paramètres reliant la variable à modéliser au vecteur d'entrées **x** = {x_k, k = 1…….p}. Ces entrées peuvent être des fonctions non paramétrées, ou à paramètres fixés, de ces variables. L'équation linéaire recherchée est de la forme :

$$g(x,\theta) = \sum_{k=1}^{q} \theta_k x_k = X\theta \tag{47}$$

Où θ = {θ_k, k=1……..p} est le vecteur des paramètres; **X**, matrice des observations de taille (n, p), est définie comme la matrice dont les éléments de la colonne k prennent pour valeurs les n mesures de la variable k. Pour chaque élément i de la base d'apprentissage, le résidu est défini comme la différence entre la valeur de la grandeur à modéliser pour cet élément i et l'estimation du modèle :

$$R_i = y^i - g(x^i,\theta) \tag{48}$$

L'apprentissage est réalisé par minimisation de la fonction de coût des moindres carrés, qui mesure l'ajustement du modèle g aux données d'apprentissage :

$$J(\theta) = \sum_{i=1}^{N} (R_i)^2 = \sum_{i=1}^{N} \left[y^i - g(x^i,\theta) \right]^2 = \|y - X\theta\|^2 \tag{49}$$

La fonction J(θ) est une fonction positive quadratique en θ: son minimum est unique. Il est donné par :

$$\boldsymbol{\theta}_{mc} = \left(\boldsymbol{X}^T \boldsymbol{X}\right)^{-1} \boldsymbol{X}^T \boldsymbol{y} \qquad (50)$$

Les paramètres θ_k sont appelés coefficients de régression partielle ; chacun d'eux mesure l'effet de la variable explicative x_k concernée sur la propriété modélisée lorsque les autres variables explicatives sont maintenues constantes.

La régression linéaire est facile à mettre en œuvre, et les coefficients θ_k obtenus peuvent être interprétés : ils mesurent l'influence de chacune des variables sur les grandeurs étudiées.

Cependant, il est souvent nécessaire d'avoir recours à des modèles de plus grande complexité.

IV-3 Réseaux de neurones artificiels [85-88]

Les réseaux de neurones formels **[85]** étaient, à l'origine, une tentative de modélisation mathématique des systèmes nerveux, initiée dès 1943 par McCulloch et Pitts **[86].**

Un ***neurone formel*** est une fonction non linéaire paramétrée, à valeurs bornées, de variables réelles. Le plus souvent, les neurones formels réalisent une combinaison linéaire des entrées reçues, puis appliquent à cette valeur une « fonction d'activation » *f*, généralement non linéaire. La valeur obtenue *y* est la sortie du neurone. Un neurone formel est ainsi représenté sur la Figure 8.

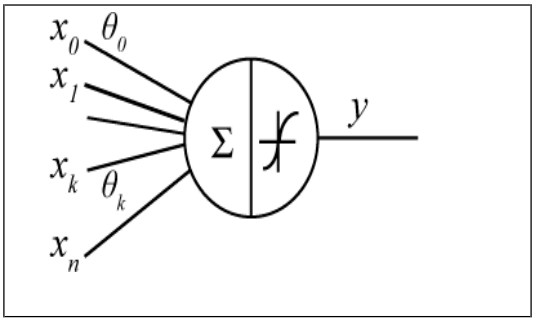

Figure 8 : Représentation d'un neurone formel

Les $\{x_k\}_{k=1.....n}$ sont les variables, ou ***entrées*** du neurone, et les $\{\theta_k\}_{k=0.....n}$ sont les ***paramètres***, également appelés synapses ou poids. Le paramètre θ_0 est le paramètre associé à une entrée fixée à 1, appelée biais. L'équation du neurone est donc :

$$y = f\left(\theta_0 + \sum_{k=1}^{n} \theta_k x_k\right) \qquad (46)$$

Les fonctions d'activation les plus couramment utilisées sont la fonction tangente

hyperbolique, la fonction sigmoïde et la fonction identité.

Les neurones seuls réalisent des fonctions assez simples, et c'est leurs compositions qui permettent de construire des fonctions aux propriétés particulièrement intéressantes. On appelle ainsi ***réseau de neurones*** une composition de fonctions « neurones » définies ci-dessus.

La Figure 9 représente un réseau de neurones non bouclé, organisé en couches (perceptron multicouche), qui comporte N_e variables, une couche de N_c neurones cachés, et N_s neurones de sortie

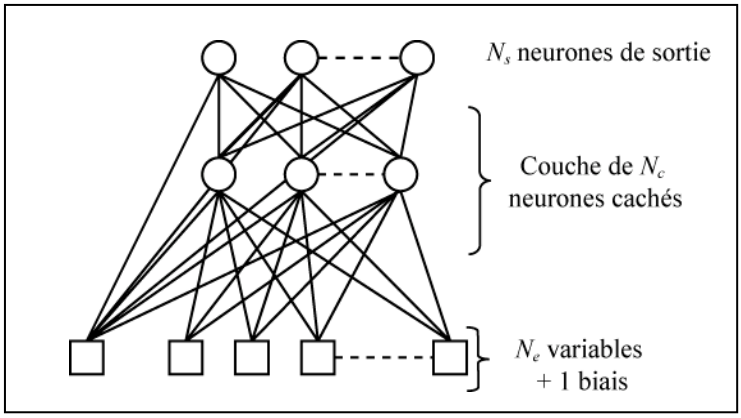

Figure 9 : Représentation d'un réseau de neurones

À chaque connexion est associé un paramètre. Les sorties du réseau sont donc des fonctions non-linéaires de ses variables et de ses paramètres. Le nombre de degrés de liberté, c'est-à- dire de paramètres ajustables, dépend du nombre de neurones de la couche cachée ; il est donc possible de faire varier la complexité du réseau en augmentant ou en diminuant le nombre de neurones cachés.

IV-3.1 Propriétés des réseaux de neurones

Les réseaux de neurones ont pour but de modéliser des processus, à partir d'exemples de couples entrées / sorties. Ils ont la propriété d'***approximation universelle*** : un réseau de neurones comportant un nombre fini de neurones cachés, de même fonction d'activation, et un neurone de sortie linéaire, est capable d'approcher uniformément, avec une précision arbitraire, toute fonction bornée suffisamment régulière, sur un domaine fini de l'espace de ses variables. De plus, il s'agit d'***approximateurs parcimonieux*** : une approximation par un réseau de neurones nécessite en général moins de paramètres que les approximateurs usuels. Le nombre de paramètres nécessaires pour obtenir une précision donnée augmente en effet linéairement avec le nombre de variables pour un réseau de neurones, alors qu'il croît exponentiellement pour un modèle linéaire par rapport aux paramètres. Cette propriété est très importante, car les réseaux de neurones demandent de ce fait moins d'exemples que d'autres approximateurs pour l'apprentissage.

IV-.3.2Apprentissage des réseaux des neurones

Considérons un ensemble d'apprentissage, constitué de n couples entrées / sorties, c'est-à-dire d'un ensemble des variables $\left\{y\left(x^i\right), i=1.....n\right\}$ et d'un ensemble de mesures de la grandeur à modéliser $\left\{y\left(x^i\right), i=1.....n\right\}$. Pour une complexité donnée, l'apprentissage s'effectue par minimisation de la fonction de coût des moindres carrés, définie par :

$$J(\theta)=\frac{1}{2}\sum_{j=1}^{N}\left[y(x^i)-g(x^i,\theta)\right]^2 \qquad (47)$$

La minimisation de cette fonction s'effectue par une descente de gradient. Cet algorithme a pour but de converger, de manière itérative, vers un minimum de la fonction de coût, à partir de valeurs initiales des poids aléatoires. À chaque étape, le gradient de la fonction est calculé, à l'aide de l'algorithme de ***rétro-propagation***. Puis les paramètres sont modifiés en fonction de ce gradient, dans la direction de la plus forte pente, vers un minimum local de *J*. Cette descente peut être effectuée suivant plusieurs méthodes : gradient simple ou méthodes du second ordre, dérivées de la méthode de Newton. Les méthodes du second ordre, généralement plus efficaces, sont les plus utilisées. La procédure de minimisation est arrêtée lorsqu'un critère est satisfait : le nombre maximal d'itérations est atteint, la variation du module du vecteur des paramètres ou du gradient de la fonction de coût est trop faible…

IV-3.3 Des réseaux de neurones particuliers : réseaux de fonctions radiales de Base

Les réseaux de neurones de fonctions radiales de base (souvent notés RBF), sont des réseaux dont la couche cachée est composée de neurones à fonction d'activation gaussienne radiale. La fonction d'activation d'un neurone *j* est par exemple :

$$f(x,\sigma_j,\theta_j)=\exp\left(-\frac{\left\|x-\theta_j\right\|^2}{2\sigma_j^2}\right) \qquad (48)$$

où σ_j est l'écart-type de la gaussienne et θ_j est le vecteur des coordonnées de son centre. Les neurones de sortie sont à fonction d'activation linéaire, et sont reliés aux neurones de la couche cachés par des poids β_j ajustables. Une sortie *y* d'un réseau à N_c neurones cachés est ainsi déterminée par :

$$y=\sum_{j=1}^{N_c}\beta_j f\left(x,\sigma_j,\theta_j\right) \qquad (49)$$

La sortie est une fonction linéaire des poids β_j et non-linéaire des paramètres des gaussiennes. Si les paramètres des gaussiennes sont fixés, la sortie devient une fonction linéaire des poids β_j, et les réseaux perdent leur propriété de parcimonie. De plus, la

réussite de l'apprentissage dépend fortement de l'initialisation. Enfin, la configuration d'un réseau de neurones RBF optimal est difficile. Lors du processus d'apprentissage du réseau, deux stratégies sont possibles. La première consiste à modifier simultanément tous les paramètres du réseau (les coordonnées des centres des fonctions radiales, leur écart-type et les poids β_j), par descente de gradient. Cependant, les dynamiques de convergence des fonctions radiales et des poids β_j sont différentes, et les poids convergent plus rapidement que les autres paramètres. L'apprentissage conduit très souvent à un minimum local. Une autre méthode consiste à optimiser séparément les paramètres de la couche cachée, par apprentissage non-supervisé, et les poids entre la couche cachée et la couche de sortie, par descente dégradaient.

Dans la plupart des applications des RNA à la chimie l'utilisation d'une seule couche cachée semble suffire **[89].** Nous avons donc utilisé dans ce travail un réseau standard à 3 couches comprenant l'entrée, la sortie et une couche cachée. L'algorithme de Levenberg-Marquardt conçu pour faciliter certains problèmes de convergence est l'un des plus utilisés pour l'apprentissage des réseaux, d'autant plus qu'il s'adapte très bien avec le choix de l'erreur quadratique moyenne comme indice de performance.

Nous avons donc utilisé l'algorithme Levenberg-Marquardt de rétropropagation (fonction TRAINLM de la boîte à outils du logiciel MATLAB 7.0 **[90]** pour l'apprentissage du réseau. Les fonctions de transfert sigmoïde (tangente hyperbolique) et linéaire ont été adoptées comme fonctions d'activation, respectivement pour les couches cachée et de sortie.

IV-4 Développement et évaluation de modèle

IV-4.1 Sélection d'un sous-ensemble de descripteurs

Des logiciels spécialisés permettent le calcul de plus de 6000 descripteurs moléculaires appartenant à différentes classes. Plutôt que de chercher à expliquer la variable dépendante (grandeur d'intérêt) par tous les régresseurs (descripteurs moléculaires), on peut chercher seulement un ensemble réduit de régresseurs qui donne une reconstitution aussi satisfaisante de la variable à expliquer. Parmi les stratégies mises en œuvre pour la sélection d'un ensemble réduit de variables explicatives, on peut citer : les méthodes de pas-à-pas (méthode descendante ; méthode ascendante et méthode dite stepwise), ainsi que les algorithmes génétiques.

En général, la comparaison se fait à l'avantage des algorithmes génétiques (AG) que nous avons appliqués dans le présent travail, et que nous rappelons succinctement.

IV-4.2 Principe de l'algorithme génétique

Dans la terminologie des algorithmes génétiques, le vecteur binaire $\underset{\sim}{I}$, appelé "chromosome", est un vecteur de dimension p où chaque position (un "gène") correspond à une valeur (1 si elle figure dans le modèle, 0 sinon). Chaque chromosome représente un modèle basé sur un ensemble de variables explicatives.

On commence par définir le paramètre statistique à optimiser (par exemple maximiser Q^2 en utilisant la validation croisée par "leave-one-out" ; (cf. infra), avec la taille P de la population du modèle (par exemple, P = 100), et le nombre maximum de variables L permises pour le modèle (par exemple, L = 10) ; le minimum de variables permises est généralement supposé égal à 1. De plus, une probabilité de croisement p_c (habituellement élevée p_c = 0,9), et une probabilité de mutation p_M (habituellement faible, p_M = 0,1) doivent être également définies.

Après définition des principaux paramètres, la mise en œuvre de l'algorithme génétique est démarrée, son évolution comprend trois étapes principales.

IV-4.3 Initialisation aléatoire du modèle.

La population est constituée au départ de modèles aléatoires avec des variables comprises entre 1 et L, puis les modèles sont ordonnés eu égard au paramètre statistique sélectionné – la qualité du modèle – (le meilleur modèle est en première position, le plus mauvais en position P) ;

IV-4.4 Etape de croisement.

A partir de la population, on sélectionne des paires de modèles (aléatoirement, ou avec une probabilité proportionnelle à leur qualité). Puis, pour chaque paire de modèles on conserve les caractéristiques communes, c'est-à-dire les variables exclues dans les 2 modèles restent exclues, et les variables intégrées dans les 2 modèles sont conservées. Pour les variables sélectionnées dans un modèle et éliminées dans l'autre, on en essaye un certain nombre au hasard que l'on compare à la probabilité de croisement p_c : si le nombre aléatoire est inférieur à la probabilité de croisement, la variable exclue est intégrée au modèle et vice versa. Finalement, le paramètre statistique du nouveau modèle est calculé : si la valeur de ce paramètre est meilleure que la plus mauvaise pour la population, le modèle est intégré à la population, à la place correspondant à son rang ; dans la cas contraire, il n'est pas pris en compte. Cette procédure est répétée pour de nombreuses paires (100 fois par exemple).

IV-4.5 Etape de mutation.

Pour chaque modèle de la population (c'est-à-dire pour chaque chromosome) p nombres aléatoires sont éprouvés, et, un à la fois, chacun est comparé à la probabilité de mutation, p_M, définie : chaque gène demeure inchangé si le nombre aléatoire correspondant excède la probabilité de mutation, dans le cas contraire, on le change de 0 à 1 ou vice versa. Les faibles valeurs de p_M permettent uniquement peu de mutations, conduisant à de nouveaux chromosomes peu différents des chromosomes générateurs.

Après obtention du modèle transformé, on en calcule le paramètre statistique : si cette valeur est meilleure que la plus mauvaise de la population, le modèle est intégré à la population, à la place correspondant à son rang ; dans le cas contraire il n'est pas pris en compte.

IV-4.6 Conditions d'arrêt.

Les étapes de croisement et de mutation sont répétées jusqu'à la rencontre d'une condition d'arrêt (par exemple un nombre maximum d'itérations défini par l'utilisateur), ou qu'il est mis fin arbitrairement au processus.

Une caractéristique importante de la sélection d'un ensemble réduit de variables par algorithme génétique est qu'on n'obtient pas nécessairement un modèle unique, mais le résultat consiste habituellement en une population de modèles acceptables ; cette caractéristique, parfois considérée comme un désavantage, fournit une opportunité pour procéder à une évaluation des relations avec la réponse à différents points de vue.

Notons que la taille du modèle est fixée par la valeur optimale de la fonction FIT de Kubinyi **[91],** calculée selon :

$$FIT = \frac{R^2}{1-R^2} \cdot \frac{n-p-1}{(n+p^2)} \qquad (51)$$

p : désignant le nombre de variables du modèle et R^2 le coefficient de détermination.

Ce critère permet de comparer entre modèles construits sur un même nombre n de données, mais avec un nombre de variables p différent.

IV-5. Paramètres d'évaluation de la qualité de l'ajustement.

Deux paramètres sont couramment utilisés :

Le coefficient de détermination multiple :

$$R^2 = 1 - \frac{SCE}{SCT} = 1 - \frac{\sum_1^n (y_i - \hat{y}_i)^2}{\sum_1^n (y_i - \bar{y})^2} \quad (52)$$

où $\hat{y}_i$ est la valeur estimée du paramètre physique, et $\bar{y}$ la moyenne des valeurs observées.

La racine de l'erreur quadratique moyenne de prédiction (désignée également par EQMP) :

$$EQMP = \sqrt{\frac{1}{n} \sum_1^n (y_i - \hat{y}_{(i)})^2} \quad (53)$$

IV-5.1 Robustesse du modèle.

La stabilité du modèle a été explorée en utilisant la "validation croisée par omission d'une observation" (LOO : cross-validation by leave-one-out) [**84**]. Elle consiste à recalculer le modèle sur (n – 1) composés de calibrage, le modèle obtenu servant alors à estimer la valeur de la propriété du composé éliminé noté $\hat{y}_{(i)}$. On répète le procédé pour chacun des n composés de l'ensemble de calibrage.

"La somme des carrés des erreurs de prédiction", désignée par l'acronyme PRESS (pour : Predictive ResidualSum of Squares) :

$$PRESS = \sum_1^n (y_i - \hat{y}_{(i)})^2 \quad (54)$$

est une mesure de la dispersion de ces estimations. On l'utilise pour définir le coefficient de prédiction :

$$Q^2_{LOO} = \frac{SCT - PRESS}{SCT} \quad (55)$$

Contrairement à R^2, qui augmente avec le nombre de paramètres de la régression, le facteur Q^2_{LOO} affiche une courbe avec maximum (ou avec palier), obtenu pour un certain nombre de variables explicatives, puis décroît par la suite de façon monotone. Ce fait confère une grande importance au coefficient Q^2_{LOO}. Une valeur de $Q^2_{LOO} > 0{,}5$ est, généralement, considérée comme satisfaisante, et une valeur supérieure à 0,9 est excellente **[92].**

IV-5.2 Domaine d'application.

Le domaine d'application a été discuté à l'aide du diagramme de Williams (traité en détail dans **[93]** représentant les résidus de prédiction standardisés en fonction des valeurs des

leviers h_i .L'équation (56) définit le levier d'un composé dans l'espace original des variables indépendantes (x_i)

$$h_i = \boldsymbol{x_i}\,(\boldsymbol{X^T X})^{-1}\boldsymbol{x_i^T} \qquad (56)$$

Où $(\boldsymbol{x_i})$ est le vecteur ligne des descripteurs du composé i et **X** (n*p) la matrice du modèle déduite des valeurs des descripteurs de l'ensemble de calibrage ; l'indice T désigne le vecteur (ou la matrice) transposé (e).

La valeur critique du levier (h*) est fixée à 3(p+1)/n. Si $h_i < h^*$, la probabilité d'accord entre les valeurs mesurée et prédite du composé i est aussi élevée que celle des composés de calibrage. Les composés avec $h_i > h^*$ renforcent le modèle quand ils appartiennent à l'ensemble de calibrage, mais auront, sinon, des valeurs prédites douteuses sans pour autant être forcément aberrantes, les résidus pouvant être bas.

IV-5.3 Test de randomisation

Ce test permet de mettre en évidence des corrélations dues au hasard. Il consiste à générer un vecteur « propriété considérée » par permutation aléatoire des composantes du vecteur réel. On calcule alors sur le vecteur obtenu (considéré comme vecteur expérimental réel) un modèle QSAR, selon la méthode habituelle. Ce procédé est répété plusieurs fois (100 dans notre cas) **[94].**

IV-5.4 Validation externe.

En plus du test de randomisation, il est intéressant **[95,96],** pour juger de la qualité du modèle, de considérer le coefficient de prédiction externe calculé comme suit :

$$Q^2ext = 1 - \frac{\sum_{i=1}^{next} (\hat{y}_{i/i} - y_i)^2 / n_{ext}}{\sum_{i=1}^{ntr} (y_i - \bar{y}_{tr})^2 / n_{tr}} \qquad (57)$$

La racine de l'écart quadratique moyen (RMSE pour Root Mean Squared Error), calculée sur différents ensembles:

- Ensemble d'estimation (appelée EQMC)
- Ensemble de prédiction externe (désignée par EQMPext).

Ces valeurs RMSE sont mieux adaptées, pour juger de la qualité d'un modèle que les valeurs de R^2 et Q^2 seules.

où n_{tr} et n_{ext} sont respectivement le nombre d'observations dans les sous ensemble de calibrage et validation et $\bar{y}_{tr}$ étant la valeur des y pour l'ensemble de calibrage

$$EQMC = \sqrt{\frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{n}} \quad (58)$$

$$EQMP_{ext} = \sqrt{\frac{\sum_{i=1}^{n_{ext}}\left(y_i - \hat{y}_i\right)^2}{n_{ext}}} \quad (59)$$

Une validation externe supplémentaire selon (Golbraikh et Tropsha, 2002) est appliquée uniquement à l'ensemble de test. Selon les critères recommandés de Tropsha *et al*, un modèle QSAR /QSPR prédictif, doit satisfaire aux conditions suivantes :

1) $Q^2_{EXT} > 0.5$ (56-a)

2) $R^2 > 0.6$ (56-b)

3) $(R^2 - R_0^2)/R^2 < 0.1$ et $0.85 < k < 1.15$ (56-c)

$(R^2 - R'^2_0)/R^2 < 0.1$ et $0.85 < k' < 1.15$ (56-d)

où

$$R = \frac{\sum(y_i - \bar{y})(\tilde{y}_i - \bar{\tilde{y}})}{\sqrt{\sum(y_i - \bar{y})^2 \sum(\tilde{y}_i - \bar{\tilde{y}})^2}} \quad (57\text{-a})$$

$$R_0^2 = 1 - \frac{\sum\left(y_i - y_i^{r_0}\right)^2}{\sum(y_i - \bar{y})^2} \quad (57\text{-b})$$

$$R'^2_0 = 1 - \frac{\sum\left(\tilde{y}_i - \tilde{y}_i^{r_0}\right)^2}{\sum(\tilde{y}_i - \bar{\tilde{y}})^2} \quad (57\text{-c})$$

$$k = \frac{\sum(y_i \tilde{y}_i)}{\sum(\tilde{y}_i)^2} \quad (57\text{-d})$$

$$k' = \frac{\sum(y_i \tilde{y}_i)}{\sum(y_i)^2} \quad (57\text{-e})$$

où R est le coefficient de corrélation entre les valeurs calculées et expérimentales dans l'ensemble de test; R_0^2 (valeurs calculées par rapport aux observées) et R'^2_0 (valeurs observées par rapport aux calculées) sont les coefficients de détermination; k et k 'sont les pentes des droites de régressions passant par l'origine pour les valeurs calculées par rapport aux valeurs observées et observées par rapport aux calculées, respectivement; $y_i^{r_0}$ et $\tilde{y}_i^{r_0}$ sont définis

respectivement par : $y_i^{r_0} = k\,\tilde{y}$ et, $\tilde{y}_i^{r_0} = k'\,y$; et les sommations sont sur tous les échantillons dans l'ensemble de test.

La raison d'utiliser et d'exiger des valeurs de k qui sont proches de 1 est que lorsque sont comparés les propriétés réelles par rapport aux prédites, un ajustement précis est nécessaire, non seulement une corrélation.