

## SequencesViewer : comment rendre accessible des motifs séquentiels de gènes trop nombreux ?

Arnaud Sallaberry \*, Nicolas Pecheur \*\*  
Sandra Bringay \*\*\*, Mathieu Roche \*\*, Maguelonne Teisseire \*\*\*\*

\* LaBRI & INRIA Bordeaux - Sud Ouest & Pikko, arnaud.sallaberry@labri.fr,

\*\*LIRMM - Université Montpellier 2, {pecheur,mathieu.roche}@lirmm.fr

\*\*\*LIRMM - Université Montpellier 3, bringay@lirmm.fr

\*\*\*\* CEMAGREF - UMR TETIS, maguelonne.teisseire@cemagref.fr

**Résumé.** Les techniques d'extraction de connaissances appliquées aux gros volumes de données, issus de l'analyse de puces ADN, permettent de découvrir des connaissances jusqu'alors inconnues. Or, ces techniques produisent de très nombreux résultats, difficilement exploitables par les experts. Nous proposons un outil dédié à l'accompagnement de ces experts dans l'appropriation et l'exploitation de ces résultats. Cet outil est basé sur trois techniques de visualisation (nuages, systèmes solaire et treemap) qui permettent aux biologistes d'appréhender de grandes quantités de motifs séquentiels (séquences ordonnées de gènes).

### 1 Introduction

Ces dernières années, les puces ADN ont été utilisées avec succès pour de nombreuses applications (diagnostic, thérapie...). Elles permettent de comparer l'expression de milliers de gènes dans différents tissus, cellules et conditions physiologiques (Hoerndli et al. (2005)). Exploiter ces données pour obtenir une interprétation biomédicale reste difficile en raison des gros volumes de données. En effet, pour une étude, les biologistes utilisent généralement moins d'une centaine de puces mais chaque puce mesure l'expression de milliers de gènes. Par exemple, les puces Affymetrix U-133 plus 2.0 mesurent 54675 valeurs numériques. Dans ce contexte, les techniques de fouille de données (Cong et al. (2004); Pensa et al. (2004)) jouent un rôle clé en permettant de découvrir des connaissances jusqu'alors inconnues. Nous utilisons ici l'algorithme DBSAP (Salle et al. (2009)) et obtenons des motifs séquentiels composés de gènes corrélés et ordonnés selon leur niveau d'expression. Selon les paramètres utilisés en entrée de DBSAP, nous obtenons entre 1.000 et 100.000 motifs pour chaque jeu de données. La quantité de résultats obtenus reste alors trop importante pour permettre aux experts de les interpréter facilement.

Outre les problèmes liés aux trop nombreux résultats, les biologistes rencontrent également des difficultés lorsqu'ils interprètent les motifs. Ils accèdent alors à des bases bibliographiques (e.g. PubMed) afin de comparer et évaluer la pertinence des corrélations découvertes. Ces interrogations sont manuelles ce qui rend le processus long et fastidieux. S'il existe désormais

des outils pour extraire automatiquement des informations à partir des données biologiques (Tanabe et al. (1999); Zeeberg et al. (2003)), ils ne sont pas adaptés aux motifs séquentiels. Dans cet article, nous proposons trois techniques de visualisation (nuages, systèmes solaire et treemap) pour permettre aux biologistes de manipuler de grandes quantités de motifs séquentiels. Ces trois modes sont adaptés à la définition des motifs et des sémantiques associées ce qui permet aux biologistes d'appréhender l'ensemble des informations "portées" par les motifs. Après avoir défini les données manipulées que nous souhaitons visualiser (Section 2), nous présentons l'outil SequencesViewer et détaillons tout particulièrement les concepts et choix réalisés (Section 3) pour assurer une ergonomie adaptée à la visualisation de nombreuses séquences ordonnées de gènes et assister les biologistes dans l'aide à la découverte de nouveautés. La section 4 dresse les perspectives associées à cette proposition.

## 2 Définitions préliminaires, Données manipulées

Les données manipulées sont des motifs séquentiels obtenus avec l'algorithme DBSAP (Salle et al. (2009)) appliqué sur des données issues de l'analyse de puces ADN. Un exemple de **séquence**  $S$  est :  $\langle (abc)(d)(fg) \rangle$ . Une séquence est composée d'**itemsets** notés entre parenthèses. Ici,  $S$  est composée de 3 itemsets :  $(abc)$ ,  $(d)$  et  $(fg)$ . Un itemset est composé d'un ou plusieurs **items** : les gènes.  $(abc)$  est composé des 3 gènes :  $a$ ,  $b$  et  $c$ . Dans un itemset, les gènes sont non ordonnés. Ils sont regroupés dans un même itemset quand leur intensité est similaire. Dans une séquence, les itemsets sont ordonnés : dans  $S$ ,  $(abc)$  est avant  $(d)$ . Les intensités des gènes  $a$ ,  $b$  et  $c$  sont moins importantes que celle du gène  $d$ . Un exemple de séquence réelle est  $\langle (MRV11)(PGAP1)(PLA2R1)(A2M)(GSK3B) \rangle$ . Cette séquence est intéressante pour les biologistes car composée de protéines dont certaines sont connues pour interférer avec les événements cellulaires de la maladie d'Alzheimer.

Une **classe** représente une population ou un sous-ensemble de cette population vérifiant une ou plusieurs propriétés communes. Dans le contexte de l'étude de la maladie d'Alzheimer, les puces peuvent être associées aux classes *sain* et *malade* (Salle et al. (2009)). Pour chaque séquence, nous calculons un **support** correspondant au nombre de cas pour lesquels une séquence est vérifiée dans une classe. Si 6 malades vérifient une séquence  $S$  dans une classe de 10 malades, alors le support de  $S$  dans la classe malade est de 6/10.

Les séquences sont organisées en **groupes**. Pour calculer la similarité entre deux séquences, nous avons utilisé la mesure S2MP (Saneifar et al. (2008)) qui prend en compte le nombre d'items communs à deux séquences ainsi que l'ordre des itemsets. Pour constituer les **groupes**, nous avons utilisé l'algorithme k-means.

Les séquences sont **résumées** et **hiérarchisées**. Par exemple, 2 séquences  $\langle (a)(b)(c) \rangle$ ,  $\langle (a)(b)(d) \rangle$  sont résumées par  $\langle (a)(b)(*) \rangle$ . Les résumés forment une hiérarchie. La séquence d'un nœud de la hiérarchie résume les séquences des fils en factorisant les itemsets communs. Nous avons utilisé l'algorithme (Nin et al. (2009)) pour générer cette hiérarchie.

Chaque séquence est associée à des documents obtenus en interrogeant Pubmed avec les noms des gènes de la séquence et les synonymes de ces noms lorsque le nombre de documents est limité (Salle et al. (2009)). Nous avons défini une mesure de proximité entre un document et une séquence qui dépend de la date de publication et du nombre de gènes évoqués dans le document. Plus un document est récent et évoque des gènes et plus ce document est proche de

la séquence.

### 3 SequencesViewer : visualisation de séquences

SequencesViewer est une application web permettant d'accéder rapidement à des informations structurées telles que nous les avons présentées précédemment. Destinée à des experts biologistes, elle fait appel à trois représentations (nuage de points, système solaire et treemap) et est dotée d'un système de navigation intuitif afin de naviguer facilement parmi celles-ci. Des captures d'écran de notre application sont disponibles en ligne <sup>1</sup>.

#### 3.1 Nuage de points

Le nuage de points (figure 1) permet de visualiser des groupes de séquences de gènes en donnant aux biologistes une première vision des centres avec leurs séquences, de la distance des centres entre eux et de la distance des séquences par rapport à leur centre.

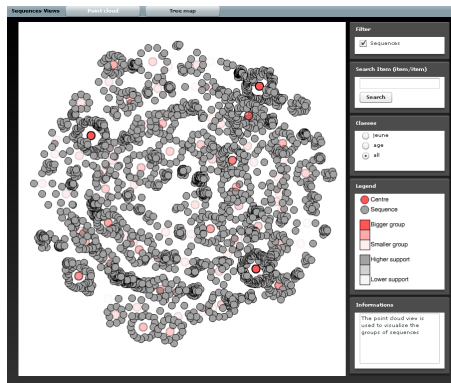


FIG. 1 – Nuage de points avec les séquences

Les centres sont placés sur le plan en fonction de la matrice des distances. Soit  $d_{ij}$  la valeur de cette matrice entre les centres  $i$  et  $j$ . Nous recherchons des coordonnées  $p_i = (x_i, y_i)$  pour chaque centre  $i$  tel que :  $\|p_i - p_j\| \approx d_{ij}$  avec  $\|p_i - p_j\|$  la distance euclidienne entre centres. Un tel problème est résoluble à l'aide d'algorithmes de *Multidimensional scaling* (Brog et Groenen (1997)) qui minimisent une fonction coût définie par :  $\sigma(p) = \sum_{i < j \leq n} \omega_{ij} (d_{ij} - \|p_i - p_j\|)^2$  avec  $\omega_{ij} = d_{ij}^{-2}$ . Pour cela, nous utilisons ici la méthode *stress majorization* (de Leeuw (1977)) et l'algorithme de (Gansner et al. (2004)) qui actualise successivement les positions  $p_i$  des centres jusqu'à ce que le système soit stable et que la fonction *stress* ne puisse plus être réduite de façon significative. L'un des inconvénients de cette méthode est de trouver un placement initial des centres. Avec un placement aléatoire, le placement final est différent à chaque exécution, l'algorithme converge moins rapidement et on peut tomber dans des minima locaux.

<sup>1</sup><http://www.labri.fr/perso/sallaber/publications/egc10/SequencesViewer.html>

Dans notre solution, nous utilisons un placement inspiré de l'algorithme *fold-free* (Priyantha et al. (2003)) que nous avons amélioré pour mieux répondre à notre problématique. Finalement, pour que les centres ne se chevauchent pas, nous utilisons l'algorithme de (Gansner et Hu (2008)).

Dans le nuage de points, l'utilisateur peut se déplacer et zoomer, afficher les informations sur une séquence dans une infobulle. Les couleurs des centres indiquent alors la quantité de séquences du groupe. Plus le centre a une couleur intense plus le groupe possède de séquences. L'utilisateur peut également affiner ses recherches à l'aide d'un filtre sur les items qui met en relief les séquences composées d'items choisis par l'utilisateur en les colorant de vert.

### 3.2 Système solaire

En double-cliquant sur l'un des centres du nuage de points, on obtient une seconde vue permettant de visualiser le groupe associé à ce centre. Le centre du groupe est placé au milieu de la fenêtre et est utilisé comme centre du repère. Les séquences se placent autour. Chacune d'entre elles est positionnée grâce à des coordonnées polaires  $(\rho_i, \theta_i)$  déterminées de la manière suivante :  $\rho_i$  est proportionnel à la distance au centre, l'angle  $\theta_i$  est calculé de façon à répartir uniformément toutes les séquences autour du centre :  $\theta_i = i \cdot \frac{2\pi}{n}$  avec  $n$  le nombre de séquences. Depuis cette vue, il est possible double-cliquer sur une séquence pour accéder à la la visualisation d'une séquence et de ses documents associés (figure 2). Comme dans la vue précédente, la séquence se place au centre de la fenêtre et les documents, représentés par des carrés, sont répartis uniformément autour à une "distance" proportionnelle à leur proximité avec la séquence. L'année de publication est représentée par le dégradé de couleur. En double-cliquant sur un document on accède à celui-ci. Cette visualisation offre une assistance à la découverte de nouveautés en facilitant l'accès aux publications ayant un lien plus ou moins fort avec les gènes examinés.

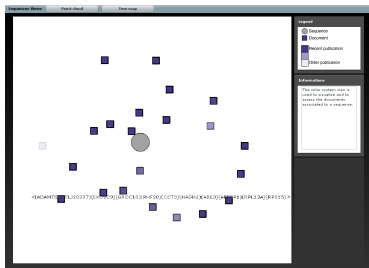


FIG. 2 – Séquence et documents associés

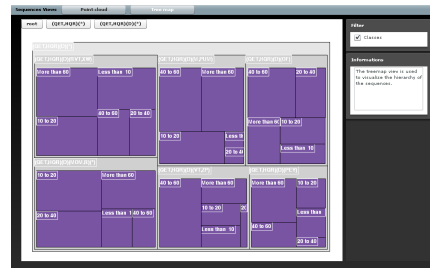


FIG. 3 – Treemap

### 3.3 Treemap

Le treemap permet une représentation efficace de larges espaces informationnels pour des données modélisées sous formes d'arbres (Johnson et Shneiderman (1991)). Nous utilisons ici un *squarified treemap*, méthode introduite par (Bruls et al. (2000)). Chaque rectangle correspond à un résumé, un fils dans la hiérarchie. Il se subdivise en deux autres rectangles dont la taille dépend de la proportion du résumé dans les classes. La navigation est très facile. Il

suffit de cliquer sur les fils pour descendre d'un niveau. Le chemin du parcours de l'utilisateur dans la hiérarchie est affiché au dessus du *treemap* et permet de revenir facilement aux niveaux supérieurs. Le *treemap* amène une vue très différente par rapport au nuage de points, il permet de faire une recherche de façon hiérarchique et volumétrique dans les séquences.

## 4 Conclusion

Nous avons proposé une approche de visualisation et de navigation au sein de gros volumes de données qui sont des séquences ordonnées de gènes. Ce travail fait suite à une collaboration étroite avec les biologistes manipulant des données issues de l'analyse de puces ADN. L'outil *SequencesViewer*, développé en partenariat avec la société PIKKO, se positionne dans une démarche d'accompagnement des experts dans l'appropriation et l'exploitation des connaissances offertes lors du processus d'extraction. L'évaluation de ce système s'est focalisée sur le nombre d'éléments (séquences, documents) adaptés pour une visualisation de qualité. Des tests ont montré que le nuage de points ne permet pas d'afficher plus de 200 centres et 2500 séquences. Le *Treemap* ne permet pas d'afficher des séquences trop grandes. De même, la présence de documents ayant un même profil (même année de publication et nombre de gènes) pose des problèmes de visualisation "système solaire". Toutefois, une revue de la littérature a montré d'une part qu'il n'existe pas à notre connaissance d'approche intégrant une visualisation hiérarchique de l'arbre des séquences (*treemap*). Les travaux associés à la visualisation de séquences dans un objectif d'alignement ne sont pas adaptés à notre contexte. D'autre part, il n'existe pas de visualisation associant séquences et documents (représenté par le système solaire). Les travaux associés à la visualisation de parties de documents ou de collections de documents ne sont pas adaptés à notre contexte. Les perspectives associées à ce travail sont nombreuses. Certaines dépendent des retours des biologistes, d'autres sont liées au caractère générique de la proposition. En effet, de nombreuses techniques de fouille de données produisent de gros volumes de résultats, parfois autant que les données initiales. Ces problèmes peuvent être solutionnés avec des approches similaires à celle proposée dans notre outil mais dans quelle mesure les choix réalisés restent-ils adaptés à des données autres que biologiques ?

## 5 Remerciements

Nous remercions la société Pikko et en particulier G. Aveline, qui nous a fourni les moyens nécessaires au bon déroulement de ce projet.

## Références

- Brog, I. et P. Groenen (1997). *Modern multidimensional scaling : Theory and applications*. New York : Springer-Verlag.
- Bruls, M., K. Huizing, et J. J. van Wijk (2000). Squarified treemaps. In *Proc. Joint Eurographics/IEEE TVCG Symp. Visualization, VisSym*, pp. 33–42.
- Cong, G. A., X. Tung, F. Pan, et J. Yang (2004). Farmer : Finding interesting rule groups in microarray datasets. In *SIGMOD Conference*, pp. 143–154.

- de Leeuw, J. (1977). Applications of convex analysis to multidimensional scaling. In *Recent developments in statistics (Proc. European Meeting Statisticians, Grenoble, 1976)*, pp. 133–145. Amsterdam : North-Holland.
- Gansner, Koren, et North (2004). Graph drawing by stress majorization. In *GDRAWING : Conference on Graph Drawing (GD)*.
- Gansner, E. R. et Y. Hu (2008). Efficient node overlap removal using a proximity stress model. In I. G. Tollis et M. Patrignani (Eds.), *Graph Drawing*, Volume 5417 of *Lecture Notes in Computer Science*, pp. 206–217. Springer.
- Hoerndli, F., D. David, et J. Götz (2005). Functional genomics meets neurodegenerative disorders. part ii : Application and data integration. *Progress Neurobiol.* 76, 169–188.
- Johnson, B. et B. Shneiderman (1991). Tree maps : A space-filling approach to the visualization of hierarchical information structures. In *IEEE Visualization*, pp. 284–291.
- Nin, J., P. Salle, S. Bringay, et M. Teisseire (2009). Using owa operators for gene sequential pattern clustering. In *22nd IEEE International Symposium on Computer-Based Medical Systems (CBMS'09)*, pp. 6.
- Pensa, R., J. Besson, et J.-F. Boulicaut (2004). A methodology for biologically relevant pattern discovery from gene expression data. In *Discovery Science, LNCS, Vol 3245*, pp. 230–241.
- Priyantha, N. B., H. Balakrishnan, E. D. Demaine, et S. J. Teller (2003). Anchor-free distributed localization in sensor networks. In I. F. Akyildiz, D. Estrin, D. E. Culler, et M. B. Srivastava (Eds.), *SenSys*, pp. 340–341. ACM.
- Salle, P., S. Bringay, et M. Teisseire (2009). Mining discriminant sequential patterns for aging brain. In *AIME '09 : Proceedings of the 12th conference on Artificial Intelligence in Medicine*, pp. 365–369.
- Saneifar, H., S. Bringay, A. Laurent, et M. Teisseire (2008). S2mp : Similarity measure for sequential patterns. In *AusDM*, pp. 95–104.
- Tanabe, L., U. Scherf, L. H. Smith, J. K. Lee, L. Hunter, et J. N. Weinstein (1999). Medminer : an internet text-mining tool for biomedical information, with application to gene expression profiling. *Biotechniques* 27, 210–4.
- Zeeberg, B. R., W. Feng, G. Wang, M. D. Wang, A. T. Fojo, M. Sunshine, S. Narasimhan, D. W. Kane, W. C. Reinhold, S. Lababidi, K. J. Bussey, J. Riss, J. C. Barrett, et J. N. Weinstein (2003). Gominer : a resource for biological interpretation of genomic and proteomic data. *Genome Biol* 4, 28.

## Summary

Techniques for extracting knowledge from huge volumes of data, obtained from DNA microarrays analysis, allow the discovery of previously unknown knowledge. However, these techniques produce many results not easily actionable by the experts. We propose a tool dedicated to the support of these experts in the process of appropriation and exploitation of these results. This tool is based on three visualization techniques (clouds, solar and treemap) that allow biologists to capture large amounts of sequential patterns (ordered sequences of genes).