Architectures parallèles pour la comparaison de séquences biologiques

D. Lavenier
IRISA / CNRS
Campus de Beaulieu
35042 Rennes cedex
lavenier@irisa.fr

Résumé

Cet article présente plusieurs machines parallèles spécialisées pour la comparaison de séquences biologiques. Ce sont des machines principalement bâties autour d'un réseau linéaire de processeurs. Leurs performances dépassent de plusieurs ordres de grandeur celles des machines programmables, permettant ainsi de faire face à l'accroissement extrêmement rapide des banques de séquences.

Abstract

This paper presents several machines dedicated to biological sequence comparison. These machines are parallel machines based primarily on linear arrays. Their performance is several orders-of-magnitude better than that of programmable machines, allowing them to face the challenge of the extremely fast growth of biological-sequence databases.

1 Introduction

Chaque année, le nombre des séquences répertoriées dans les bases de données biologiques s'accroît d'environ 50 %.

En dépit de l'augmentation constante des performances des calculateurs, l'analyse de ces données sur des machines classiques – processeurs séquentiels du commerce – devient de plus en plus conséquente. Si aujourd'hui la majorité des applications se satisfait d'un usage de ces machines, il risque d'en être tout autrement demain, lorsque les banques renfermeront des centaines de milliers (voire des millions) de séquences. Dès maintenant, des solutions informatiques nouvelles doivent être proposées pour faire face à cette progression.

Trois approches sont possibles : utiliser des *réseaux de calculateurs*, des *calculateurs massivement parallèles*, ou réaliser des *machines spécialisées*. Cet article traite plus particulièrement de cette dernière catégorie de machines, et notamment des machines systoliques spécialisées dans ce domaine.

Répartir un programme de traitement de séquences biologiques sur un réseau de calculateurs (par exemple un réseau de stations de travail) est une première méthode. Elle a été récemment testée dans le cadre de l'établissement de la cartographie physique de l'ensemble du génome humain [6] [4] et a prouvée sa validité. Cependant, cette approche est limitée par le nombre de stations de travail accessibles et disponibles – en pratique, quelques dizaines au plus.

L'usage de *calculateurs massivement parallèles* constitue une seconde approche, actuellement en pleine expansion. La parallélisation des algorithmes de comparaison de séquences sur ces machines permet de réduire fortement la durée des traitements. Certains logiciels, tels que *Blaze* ou *MPsrch*, sont déjà disponibles sur des calculateurs massivement parallèles de type MasPar [12]. Ces logiciels sont compétitifs par rapport à des programmes similaires – Fasta [11] [13] ou Blast [1] – optimisés pour des machines séquentielles : à temps de calcul comparables, les résultats des machines parallèles semblent de meilleure qualité.

Cependant, certains traitements relatifs à l'analyse ou la structuration de banques entières [17] conduisent, même en utilisant ces machines, à des temps de calcul gigantesques. A titre d'exemple, l'extraction de sous-séquences homologues, c'est à dire de portion de séquences similaire, dans une banque de 20 000 séquences protéiques (Swiss-Prot 21, par exemple) demande approximativement quatre jours sur une machine MasPar MP-1 à 1024 processeurs, lorsqu'une méthode rigoureuse [16] est appliquée.

La complexité de ces algorithmes est directement proportionnel au nombre de séquences des banques. Ainsi, l'analyse d'une banque de 200 000 séquences protéiques (taille estimée de la banque Pir vers 1996 ou 1997) demanderait 400 jours. En tablant sur une machine plus rapide – ou ayant plus de processeurs – on peut bien sûr réduire ce temps, mais il est clair que celui-ci restera très important et donc un facteur limitatif d'investigation.

On peut améliorer les performances d'un traitement informatique d'un ordre de grandeur en faisant appel à des *architectures spécialisées*, parallèles ou non. En spécialisant les processeurs, on peut gagner aisément un facteur 10 à 100 sur la vitesse des traitements. De plus, les machines parallèles spécialisées sont moins encombrantes ; il est donc possible de gagner aussi un ordre de grandeur sur les performances en augmentant le nombre de processeurs.

2 Les algorithmes de comparaison de séquences

On entend par algorithme de comparaison de séquences les algorithmes qui permettent de résoudre – entre autre – les problèmes suivants :

- recherche de motifs : on dispose d'un jeu de motifs (courtes séquences particulières) et on détermine s'ils appartiennent à une ou plusieurs séquences ;
- recherche de profils : on établit un squelette de séquence et on recherche celles qui s'en rapprochent le plus ;

- comparaison de séquences : on évalue le degré de ressemblance entre deux ou plusieurs séquences ;
- recherche de segments homologues : on localise des régions qui présentent des ressemblances.

Pour chacun de ces problèmes, il existe des méthodes plus ou moins strictes pour évaluer les ressemblances. Elles vont d'une correspondance exacte à une correspondance approximative. Ces dernières sont, en général, très coûteuses en calculs, mais donnent de meilleurs résultats ; elles font appel à des méthodes de programmation dynamique dont la complexité est proportionnelle au carré des tailles des séquences.

C'est dans cette dernière catégorie d'algorithmes que les machines spécialisées sont exploitées. En effet, ces algorithmes présentent certaines caractéristiques – comme la régularité – qui se prêtent extrêmement bien à une mise en œuvre sur des machines parallèles.

A titre d'exemple, nous présentons un algorithme très utilisé et représentatif des complexités mises en jeu ; il a été proposé par Smith et Waterman [16] pour déterminer des sous-séquences homologues entre deux séquences biologiques. Cet algorithme permet d'identifier des portions de séquences similaires en prenant en compte les erreurs de substitution et les erreurs d'insertion/omission (ou gap) multiples.

A partir de 2 séquences $S1$ et $S2$ de longueur respective $l1$ et $l2$, l'algorithme calcule une matrice de valeurs H de taille $l1 \times l2$; Chaque valeur $H(i,j)$ représente une vraissemblance locale et est déterminée par la relation de récurrence suivante :

$$H(i,j) = Max \begin{cases} 0 \\ Max_{1 \leq k \leq i}(H(i-k,j) - g_k) \\ Max_{1 \leq l \leq j}(H(i,j-l) - g_l) \\ H(i-1,j-1) + sub(S1_i, S2_j) \end{cases} \quad (1)$$

avec les initialisations : $H(i,0) = 0 \quad (0 \leq i \leq l1)$ et $H(0,j) = 0 \quad (0 \leq j \leq l2)$

$sub(S1_i, S2_j)$ représente le coût de substitution du caractère $S1_i$ par le caractère $S2_j$. g_k représente le coût d'un gap de k caractères ; il est déterminé par : $g_k = \alpha + (\beta - 1)$, α étant le coût du premier gap et β le coût des suivants.

Lorsque la matrice a été calculée, l'examen de ces composantes permet de détecter les endroits où des similarités locales apparaissent. Une procédure de *backtrack* permet ensuite d'identifier complètement les sous-séquences homologues à partir du maximum local trouvé.

En fait, la complexité de cet algorithme – et plus généralement la complexité des algorithmes du même type – est surtout fonction de la première étape. En effet, si on a pris soin, au cours de celle-ci, de mémoriser les maximum locaux (s'ils existent et s'ils sont significatifs), la phase suivante qui récupère les sous-séquences similaires a un temps de calcul très petit (voire négligeable) devant l'établissement de la matrice. Dans la pratique, lorsqu'une séquence est comparée à une base de données, seules quelques séquences présentent des homologies locales potentielles. La deuxième phase est alors extrêmement réduite.

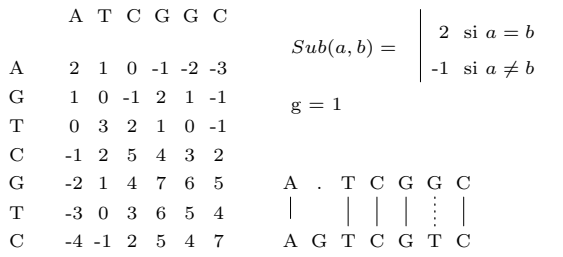

Figure 1: calcul de la distance entre AGTCGTC et ATCGGC

3 Parallélisation

Ce paragraphe a pour but de montrer comment se parallélise les algorithmes de comparaison de séquences biologiques. Pour simplifier, considérons la relation de récurrence suivante :

$$D(i,j) = Max \begin{cases} D(i-1,j) - g & 1 \leq i \leq l_1 \\ D(i,j-1) - g & 1 \leq j \leq l_2 \\ D(i-1,j-1) + sub(S1_i, S2_j) \end{cases} \quad (2)$$

qui calcule une distance entre deux séquences $S1$ et $S2$ de longueur respective l_1 et l_2. Cette distance est donnée par $D(l_1, l_2)$, calculée récursivement à partir des initialisations :

$$D(0,0) = 0$$

$$D(i,0) = D(i-1,0) - g$$

$$D(0,j) - D(0.j-1) - g$$

Cet exemple simple est représentatif de la majorité des algorithmes de comparaison de séquences qui utilisent des techniques de programmation dynamique. Si on parvient à le paralléliser sur une architecture adéquate, il en sera de même pour l'ensemble des autres.

La figure 3 donne une représentation graphique, sur un exemple, du calcul de la distance des séquences $S1$=`AGTCGTC` et $S2$=`ATCGGC`. Le coût d'un gap est de 1 et le coût de substitution d'un nucléotide a par un nucléotide b est donné par :

$$Sub(a,b) = \begin{cases} 2 & si \quad a = b \\ -1 & si \quad a \neq b \end{cases}$$

La distance résultante est de 7. Elle équivaut à 5 mises en correspondance exacte entre nucléotides, une subsitution et un gap. L'alignement des deux séquence peut se représenter par :

```
A G T C G T C
|   | | | : |
A . T C G G C
```

Sur une machine séquentielle, il faut N^2 pas de calcul pour comparer deux séquences de longueur N, un pas étant relatif au calcul d'une récurrence. La parallélisation a pour but de réduire ce nombre de pas en distribuant les calculs sur des processeurs différents.

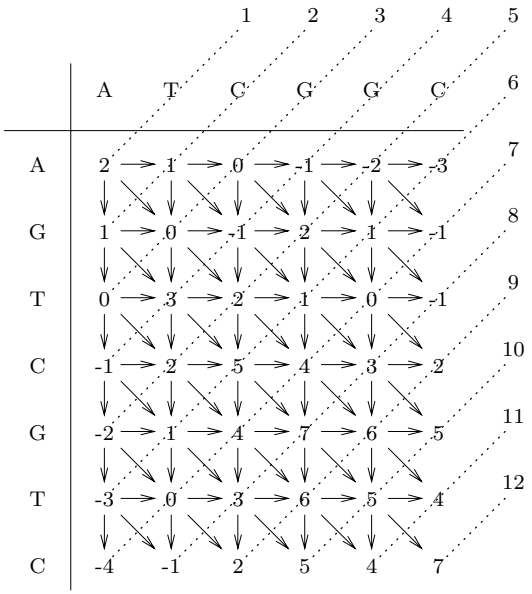

Figure 2: graphe de dépendance

Cette répartition ne peut se faire qu'en accord avec le graphe de dépendance associé au calcul. Toujours sur le même exemple, la figure 2 indique ces dépendances (traits plein). Elle indique également les instants auxquels les calculs peuvent avoir lieu (pointillé). Ainsi, au temps 1, seul le calcul relatif à $D(1,1)$ (cf équation 2) peut être réalisé. Au temps 2, les valeurs ($D1,2$) et $D(2,1)$ peuvent être calculées, au temps 3 les valeurs $D(1,3)$, $D(2,2)$ et $D(3,1)$, etc.

Plus généralement, au temps k l'ensemble des $D(i,j)$ tels que $k = i + j - 1$ peuvent être calculées. On en déduit qu'en $2N - 1$ pas de calcul le calcul complet peut être réalisé, si l'on dispose, toutfois, d'une structure de machine appropriée.

En fait, les réseaux systoliques sont idéals pour supporter ce type de parallélisation. Le concept, dû à Kung [10], consiste à obtenir, pour résoudre un problème donné, une architecture régulière, faite de processeurs simples et identiques, connectés suivant une topologie régulière et locale (pas de liens éloignés entre processeurs) où les données circulent à vitesse constante. L'ensemble fonctionne par battements d'où le nom donné à ce type de machine. Un battement est appelé cycle systolique.

L'idée de base est, ici, d'associer un processeur au calcul de chaque distance $D(i,j)$. On obtient alors une matrice de processeurs de taille $N \times N$. Dans le cas des séquences biologiques dont la longueur moyenne est d'environ 300 pour les chaînes protéique et 1000 pour les séquences d'ADN, on s'aperçoit très vite que, pratiquement, un tel réseau (1 million de processeurs) est irréalisable.

En fait, l'analyse du graphe de dépendance montre que les calculs s'effectuent sur une seule diagonale à un instant donné. Autrement dit, si l'on considère la matrice de processeurs, un seul processeur par colonne – ou par ligne – est actif à chaque cycle systolique.

Cette remarque permet de réduire le réseau bidimensionnel en un réseau linéaire en considérant qu'une colonne – ou une ligne – de processeurs peut être émulée par un seul processeur. Ainsi, si on choisit une projection verticale (émulation des colonnes), au temps 1, le premier processeur calculera la valeur de $D(1,1)$, puis au temps 2 la valeur de $D(2,1)$, au temps 3 la valeur de $D(3,1)$, etc.

Le temps de comparaison sur une structure linéaire est identique à celui d'une architecture bidimensionnelle, à savoir $2N - 1$ cycles systoliques. Dans cette dernière approche, le nombre de processeurs est égale à la longueur d'une séquence. Ce nombre est raisonnable et autorise la réalisation d'une machine.

4 Réseaux systoliques spécialisés dans la comparaison des séquences biologiques

L'architecture de ces machines, représentée schématiquement sur la figure 3, se compose d'un réseau de N processeurs connectés de voisin à voisin, d'un ordinateur hôte et d'une interface réalisant la liaison entre les deux. L'application principale se déroule sur l'ordinateur hôte. Elle sollicite le réseau de processeurs lorsque des calculs intensifs sont requis. La majorité des machines spécialisées, étudiées ou réalisées à ce jour, possèdent ce type de structure.

Typiquement, la comparaison de séquences s'effectue de la manière suivante : la séquence à tester est répartie à raison d'un caractère par processeur et la banque est injectée à une extrémité du réseau. Les éléments de la banque circulent à travers le réseau et rencontrent ainsi tous les caractères de la séquence ; à chaque rencontre un calcul matriciel élémentaire – un pas de la relation de récurrence – est réalisé.

Les performances des machines dépendent à la fois de la complexité des processeurs (ou complexité du calcul à réaliser), du nombre de processeurs et de la fréquence à laquelle la base de donnée est émise vers le réseau. Les performances maximales (P_{max}) d'une machine, exprimées en nombre de calculs matriciels élémentaires par seconde (CME/s), sont données par la formule :

$$P_{max} = f \times N$$

où f est le nombre de caractères émis par seconde vers le réseau et N le nombre de processeurs.

Les performances réelles d'une machine sont en général moindres. Dans la réalité, il est difficile d'utiliser de manière optimale le réseau de processeurs : la taille d'une séquence test (T) est rarement égale à la taille du réseau (N) ; si elle est plus courte, $(N - T)$ processeurs sont inactifs. Dans ce cas, la quantité de calculs effectués est :

$$f \times (N - T)$$

Si la séquence est plus longue ($T = N \times a + b$), le traitement doit être découpé en $a + 1$ étapes, les a premières étapes utilisant pleinement le réseau et la dernière ne l'utilisant que partiellement.

Plusieurs machines ont été bâties sur ce modèle. Les paragraphes suivants décrivent rapidement cinq de machines. Elles ont toutes pour vocation d'accélérer les recherches effectuées sur les banques de séquences biologiques. Cette description n'est pas exhaustive. Elle est cependant représentative de l'état de l'art actuel.

La machine Bisp

La machine Bisp [5] (*Biological Information Signal Processor*), développée à l'Institut de Technologie de

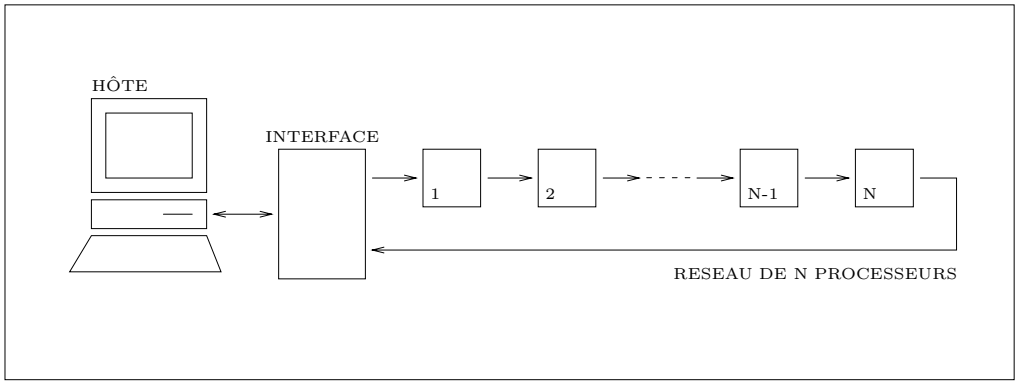

Figure 3: architecture typique d'une machine spécialisée dans l'analyse de séquences

Californie, est constituée d'un réseau systolique linéaire spécialisé dans la recherche d'alignements locaux [16]. Elle utilise une puce spécialisée dans laquelle 16 processeurs sont intégrés. Au maximum, 262 144 puces peuvent être cascadées pour former un réseau de 4 194 304 processeurs (!). Un prototype de 256 processeurs (un circuit imprimé de 16 puces) a été réalisé et testé.

Les performances de la machine dépendent bien sûr du nombre de processeurs. La fréquence d'horloge du circuit étant de 12,5 Mhz et un processeur étant capable d'effectuer un calcul par période, la puissance de calcul est de $12,5 \times 10^6 \times N$ CME/s. Une machine Bisp de 256 processeurs possède une puissance de calcul maximale de $3,2 \times 10^9$ CME/s.

La machine BioScan

La machine BioScan [15] est, comme la précédente machine, un système dédié permettant l'accélération d'un algorithme particulier. Il s'agit cependant d'un algorithme moins complexe puisque la recherche d'homologie porte sur des segments de même longueur sans possibilité d'insertion ni d'omission. Cette machine est développée par l'université de Caroline du Nord à Chapel Hill (Etats-Unis).

BioScan est une architecture linéaire de faible complexité. Chaque puce contient 812 processeurs 1 bit et la machine contient 16 puces, soit un ensemble de 12 992 processeurs. Le nombre élevé de processeurs lui confère une grande puissance de calcul. La fréquence d'horloge de la puce est de 32 MHz. À raison de 16 cycles d'horloge pour un calcul matriciel élémentaire, BioScan atteint 25×10^9 CME/s. Rappelons toutefois que c'est un calcul beaucoup plus simple que dans le cas précédent et que l'unité CME n'a pas exactement la même signification.

La machine Bioccelerator

La machine Bioccelerator [7], développée au Weizmann Institute of Science, en Israël, est une machine spécialisée dans l'accélération de certains programmes du progiciel GCG [8] (Genetics Computer Group). Le cœur de la machine est constitué de circuits logiques reconfigurables [14] (FPGA : *Field Programmable Gate Array*).

Cette machine n'est donc pas dédiée à un algorithme particulier : en modifiant les configurations associés aux FPGA, les structures de la machine peuvent être directement adaptées à un algorithme donné. Par contre, le temps d'implantation et de mise au point d'un nouvel algorithme requiert les compétences d'un architecte de machines.

Bioccelerator est commercialisée ; dans sa première version, elle accélère le programme *ProfileSearch* du progiciel GCG. Les performances mesurées sur ce programme particulier sont de 320×10^6 CME/s dans sa configuration maximale.

La machine Splash-2

La machine Splash-2 [2], conçue au SRC (*Supercomputing Research Center - Institut for Defense Analyses*), présente d'intéressantes possibilités. C'est également un système à base de circuits programmables (FPGA). Elle est le successeur de la machine Splash-1 initialement étudiée pour des applications en biologie moléculaire. Splash-2 est un réseau linéaire dont chaque nœud est composé d'un FPGA (Xilinx 4010, l'équivalent de 10 000 portes logiques) et d'une mémoire statique de 512 Ko. La configuration maximale est de 256 nœuds répartis sur 16 cartes distinctes.

Les premières mise en œuvre d'algorithmes d'analyse de séquences [9] montrent qu'il est possible d'implanter plus d'un processeur par nœud. On peut estimer à deux le nombre de processeurs supportant l'algorithme de Smith et Waterman et pouvant être contenus dans un nœud, soit un réseau de 512 processeurs. Avec une fréquence d'horloge de 20 MHz, la machine atteint 10×10^9 CME/s.

La machine Samba

La machine Samba (*Systolic Accelerator for Molecular Biology Application*) est un projet d'architecture pour l'analyse des banques de séquences développé à l'Irisa. Samba est une architecture parallèle linéaire. Elle est constituée de puces spécialisées pour la partie réseau et de FPGAs pour la partie contrôle et alimentation en données. Une machine prototype de 256 processeurs est prévue. Les performances du réseau se situent donc aux alentours de 5.12×10^9 CME/s ($256 \times 20 \times 10^6$).

Le lien avec une station de travail est réalisé (dans la version prototype) par une carte expérimentale à base de circuits logiques reconfigurables (FPGA), la carte Perle1 développée à DEC-PRL [3].

Cette approche mixte (puces spécialisées/FPGAs) est intéressante. Elle permet d'exploiter la régularité de la structure linéaire en intégrant plusieurs

processeurs par puce. On obtient ainsi un réseau de grande taille dans un minimum de volume. D'autre part, la technologie des circuits logiques reconfigurables permet d'optimiser l'interfaçage entre l'ordinateur hôte et le réseau sans le figer complètement. Suivant les problèmes à résoudre (comparaison de deux séquences, d'une séquence avec une banque, de deux banques), le contrôle du réseau exige des mises en œuvre différentes qui, pour être efficaces, doivent être matériellement adaptées.

5 Conclusion

Les algorithmes couramment utilisés pour l'analyse des banques de séquences se prêtent particulièrement bien à une parallélisation sur des réseaux linéaires de processeurs. Ces algorithmes étant coûteux, en terme de calcul, des machines parallèles spécialisées sont indispensables pour faire face à l'accroissement extrêmement rapide des banques de séquences.

Plusieurs machines, principalement bâties autour de structures linéaires, ont d'ores et déjà été étudiées et développées. Certaines sont même opérationnelles comme la machine BioScan (connectée à un serveur accessible par réseau) ou la machine Bioccelerator (commercialisée par la société Compugen, Israël).

Ces machines sont conçues autour de puces spécialisées et/ou de circuits logiques reconfigurables (FPGAs). L'intégration de plusieurs processeurs par puce permet d'obtenir des machines de faible dimension (un circuit imprimé). Connectées à une station de travail standard, elles sont nettement plus performantes que les machines parallèles programmables.

Ces machines visent principalement deux secteurs d'utilisation. Le premier concerne la mise à disposition d'une ressource performante à une communauté de personnes ayant des besoins ponctuels. On peut imaginer une connexion à un serveur offrant immédiatement des services impossibles à réaliser localement. Dans ce cas, le centre serveur dispose d'une ou plusieurs machines spécialisées qu'il active en fonction des requêtes.

L'autre secteur d'utilisation concerne l'usage intensif de telles machines. Les recherches menées, par exemple, sur la classification d'une banque de séquences par analyse d'homologies [17] demandent une masse de calculs gigantesque. Dans ce cas, l'exploitation permanente (ou sur un temps très long) d'une ressource d'un centre serveur n'est pas envisageable. Le faible coût d'une machine spécialisée (comparativement à une machine parallèle programmable qui offrirait les mêmes performances) permet à une équipe de recherche de se doter d'une telle machine, voire de plusieurs.

References

[1] S.F. Altschul, W. Gish, W. Miller, E.W. Myers, and D.J. Lipman. Basic local alignment search tool. *J. Biol. Mol*, (215):403–410, 1990.

[2] J.M. Arnold, D.A. Buell, and E. G. Davis. SPLASH 2. In *4th Annual ACM Symposium on Parallel Algorithms and Architecture*, 1992.

[3] P. Bertin. *Mémoires actives programmables : conception, réalisation et programmation.* PhD thesis, Université Paris 7, juin 1993.

[4] C. Bellann -Chantelot and all. Mapping the whole human genome by fingerprint yeast artificial chromosomes. *Cell*, 70:1059–1068, 1992.

[5] E. Chow, T. Hunkapiller, and J. Peterson. Biological Information Signal Processor. In *ASAP*, pages 144–160, sep 1991.

[6] JJ. Codani and B. Lacroix. *Computational aspect of genome physical mapping.* research report 1560, INRIA, dec 1991.

[7] Compugen. *The BIOCCELERATOR machine.* Israel, 1993.

[8] *Program Manual for the GCG package, version 7.* Genetic Computer Group, 575 Science Drive, Madison, Wisconsin, USA 53711, Apr 1993.

[9] D. T. Hoang. Searching Genetic DataBases on SPLASH-2. In D.A. Buell and K.L. Pocek, editors, *FPGAs for custom computing machines*, pages 185–191, IEEE Computer Society Press, apr 1993.

[10] H.T. Kung. Why Systolic Architectures? *Computer*, 15(1):37–46, 1982.

[11] R.J. Lipman and W.R. Pearson. Rapid and sensitive protein similarity searches. *Science*, 227:1435–1441, 1985.

[12] J. R. Nickolls. The Design of the MasPar MP-1 : A Cost Effective Massively Parallel Computer. In *COMPCON*, IEEE, feb 1990.

[13] W. R. Pearson and D.J. Lipman. Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci.*, 85:3244–3248, 1988.

[14] J. Rose, A. El Gamal, and A. Sangiovanni Vincentelli. Architecture of Field-Programmable Gate Arrays. *Proceedings of the IEEE*, 81(7):1013–1029, jul 1993.

[15] R.K. Singh, S.G. Tell, C.T. White, D. Hoffman, V.L. Chi, and B.W. Erickson. A Scalable Systolic Multiprocessor System for Analysis of Biological Sequences. In G. Borrielo and C. Ebeling, editors, *Research on Integrated Systems*, pages 168–182, 1993.

[16] T.F. Smith and M.S. Waterman. Identification of common molecular subsequences. *J. Mol. Biol*, (147):195–197, 1981.

[17] E. L. Sonnhammer and D. Kahn. The Modular Arrangement of Proteins as Inferred from Analysis of Homology. *Protein Science*, 3:482–492, 1994.