Entwurf: Version 4

SP^3: A Workflow Management System for parallel Sequence Pre-Processing

(Arbeitstitel)

Steffen Andrew Scheer
Betreuer: Prof. Dr. Ulf Leser

16. März 2008

1 Einführung

Die Entwicklung und Verbesserung der DNA-Analysetechniken ist seit mehr als zehn Jahren ein intensives Forschungsgebiet und hat einen erheblichen Einfluss auf die biologische und medizinische Forschung genommen. Durch die stetige Verbesserung der Sequenziertechnologie ist man gegenwärtig in der Lage, innerhalb von 24 Stunden mit einem einzelnen Gerät eine Sequenzlänge von bis zu 100 Megabasen zu lesen. Diese Generation der schnelleren und preiswerteren Methoden [Mar08] [GJB06] verspricht ein sehr breites Feld der genetischen Analysen zu adressieren. Darunter fallen: die vergleichende Genomanalyse, das Transkriptomprofiling sowie die Hochdurchsatz SNP-Identifikation.

Die Beurteilung der Sequenzqualität nach unterschiedlichen Kriterien ist von großer Bedeutung, um Fehler bei dem Sequenzclustering und Contigassembly zu minimieren. Aus diesem Grund ist es notwendig, reproduzierbare und einfach interpretierbare Bewertungen der Sequenzqualität zu generieren. Anhand dieser kann situativ entschieden werden, ob eine Basensequenz verworfen wird oder nicht.

Sequenzrohdaten durchlaufen neben der Projektintegration, bei der die Probe dem entsprechenden Projekt zugeordnet und die Basenabfolge bestimmt wird, noch mindestens vier weitere Qualitätskontrollen. Dazu gehört die Sequenzqualitätsbewertung, die Vektorbeschneidung, das Maskieren von „Low-complexity“ Bereichen und die Kontaminationsuntersuchung. Für jeden Arbeitsschritt existieren spezielle Programme, die je nach Zielstellung unterschiedlich zusammengestellt und parametrisiert werden müssen. Die Vielfalt an Werkzeugen, die auf den Daten operieren und die Mentalität der Entwickler, ihr eigenes Datenformat als defakto Standard zu erheben, macht eine Kombination aufwendig, da viele Datentransformationen durchgeführt werden müssen. Die Herausforderung besteht nun darin, diese Programme so flexibel miteinander verbinden zu können, dass eine performante und projektspezifische Arbeitsumgebung für die Analyse von Hochdurchsatzsequenzen mit wenig Aufwand erreicht werden kann.

Die sich stetig verbessernden Verfahren und die damit verbundene Automatisierung der Rohdatengenerierung stellt anwendende Institutionen zunehmend vor Probleme, die Daten zu organisieren und zeitnah zu verarbeiten. Bis heute ist es nicht unüblich eine Qualitätsbewertung der Rohdaten semiautomatisch durch die Kombination von mehreren Programmen durchzuführen, die nacheinander manuell ausgeführt werden, oft noch verbunden mit diversen Formatkonvertierungen. Diese Methoden sind bei zunehmendem Projektumfang und Durchsatz nicht mehr adäquat, insbesondere unter den Gesichtgesichtspunkten der Datenintegrität und der Reproduzierbarkeit.

Die Datenmenge, im Detail die Anzahl der Sequenzen, hat mittlerweile einen kritischen Punkt erreicht, bei dem die Laufzeit einer Pipeline im hohen Maße durch die I/O Operationen beeinflusst

wird, insbesondere wenn sich die Daten auf einem Fileserver befinden. Effizientere Speichermöglichkeiten im Sinne von Reduktion der Dateianzahl bei schnelleren Zugriffszeiten müssen konsequent umgesetzt werden, um dieses Problem zu lösen. Erste Ansätze kann man beispielsweise bei der Assembly und Finishing Software Consed [Gor06] oder bei den 454 Sequenziergeräten[1] beobachten. Beide verwenden ein Archivformat, um die Dateianzahl zu minimieren. Allerdings sind beide Implementierungen adhoc Lösungen für spezielle Sequenzformate und nicht für einen wahlfreien Zugriff optimiert.

2 Abgrenzung

In der Literatur sind einige Systeme beschrieben, die eine integrierte Pipeline bieten, in der Sequenzrohdaten nach einem vorgegebene Arbeitsablauf verarbeitet werden können. Die Beschränkung auf spezielle Arten von Sequenzdaten [DAC05] oder die starren Arbeitsabläufe sowie die Festlegung auf bestimmte Programme für deren Umsetzung [ALV06] machen diese Systeme für einen generellen Einsatz unbrauchbar. Die Bioinformatik ist ein sehr agiles Forschungfeld und verbesserte Algorithmen und Programme werden regelmäßig veröffentlicht. Vor diesem Hintergrund sollte ein Systeme vom Entwurf her möglichst offen sein.

Wissenschaftliche Workflows sind ein recht neues Verfahren, komplexe wissenschaftliche Arbeitsabläufe zu beschreiben [LB05]. Durch die Fokussierung auf die Datentransformation und den Datenfluss zwischen den einzelnen Arbeitsschritten (Aktoren) lassen sich komplexe Arbeitsabläufe modellieren und durch ein Workflow Management System (WMS) [LAB06] [OAF04] ausführen. Die Realisierung der einzelnen Arbeitsschritte als Webservice und die Orchestrierung durch ein WMS ist allerdings vor den Gesichtspunkten des zu übertragenden Datenvolumens, einer fehlenden Lastverteilung und Parallelisierung nicht praktikabel. Für die vorliegende Aufgabe benötigt man ein WMS, das anhand der zu verarbeitenden Datenmenge die Ausführung eines Arbeitsablaufes in mehrere Prozesse aufteilt und auf zur Verfügung stehenden Ressourcen im lokalen Netz verteilt. Der Einsatz einer Clustersoftware für ein verteiltes Ressourcen Management (DRM) scheint sinnvoll zu sein. Ein DRM System bietet Funktionalität im Bereich der Lastverteilung, Verwaltung von Rechenaufträgen sowie der Überwachung dieser. Man kann für jedes verwendete Programm eine optimale Ausführungsumgebung wählen - von paralleler Stapelverarbeitung bei datenverarbeitungsintensiven Prozessen über eine parallele Ausführung auf Muliprozessor Architekturen via Threads und Shared Memory bis zu einer parallelen Ausführung auf mehren Knoten basierend auf einer MPI[2] oder PVM[3] Implementation.

3 Zielsetzung

In dieser Arbeit wird der Entwurf und die Implementation einer Arbeitsumgebung für die parallele Verarbeitung von Sequenzdaten entwickelt sowie ein Mechanismus zum Indizieren der unterstützten Sequenzdatenformate. Ziel ist es zum einen, eine Workflowsprache zu entwerfen mit der es möglich ist, die gängigen Werkzeuge der Bioinformatik zu parametrisieren und durch einen Interpreter verteilt auszuführen. Zum anderen soll die Datenhaltung auf Dateisystemebene optimiert werden indem alle Sequenzdaten archiviert gespeichert werden. Für die Repräsentation der Ergebnisse wird ein XML-Datenschema entwickelt.

4 Umsetzung

Die Entwicklung erfolgt im wesentlichen in drei Schritten. **(1)** Zunächst müssen Datenschemata für die Konfiguration eines Pre-Processing Workflows und die Repräsentation der Resultate definiert

[1] Genome Sequencer FLX System - https://www.roche-applied-science.com/sis/sequencing/flx/index.jsp
[2] The Message Passing Interface standard - http://www-unix.mcs.anl.gov/mpi
[3] Parallel Virtual Machine - http://www.csm.ornl.gov/pvm

werden. **(2)** Im nächsten Schritt gilt es Strategien zu entwerfen, mit denen man die Sequenzrohdaten in ihrem ursprünglichen Format archiviert und effizient darauf zugreift. **(3)** Im letzten Schritt wird die parallele Ausführung eines Workflows auf einem Cluster geplant und umgesetzt.

4.1 Datenformate

Möchte man Sequenzdaten von verschiedenen Geräteherstellern verarbeiten, sieht man sich mit einer Vielzahl an verschiedenen Formaten konfrontiert. Hinzu kommen noch die verschiedenen Ein- und Ausgabeformate der verwendeten Werkzeuge. Diese Tatsache macht eine Kombination schwierig, da keine einheitliche Datenstruktur existiert. Die Idee besteht nun darin, ein XML-Datenschema zu entwerfen, dass eine allgemeine textuelle Repräsentation der Sequenzdaten ermöglicht. Aus dieser können alle notwendigen Formate erzeugt und die Resultate der verschiedenen Programme gespeichert werden. Der Entwurf basiert lose auf dem Schema von NCBI[4].

Für die Repräsentation eines Workflows wird ebenfalls ein XML-Datenschema verwendet. Ein Workflow besteht aus einer Liste von Elementen, die jeweils eine Parametrisierung des verwendeten Programms beschreibt.

4.2 Datenhaltung und Zugriff

Je nach Größe eines Geräteparks und dem Durchsatz kann die Anzahl der täglich zu verarbeitenden DNA-Sequenzen bei mehreren hundertausend liegen. Um die Anzahl der Dateien im Dateisystem möglichst niedrig zu halten, werden alle Sequenzdaten in ihrem ursprünglichen Format archiviert. Der Zugriff erfolgt über eine zusätzliche Indexdatei, die den Probennamen der Sequenz und die Startposition der Datei im Archiv enthält. So ist es möglich, beliebige unkomprimierte Archivformate zu indizieren, wie z.B. TAR[5] oder SFF[6]. Diese Technik eignet sich auch, um Elemente in einer XML Datei zu indizieren. Auf diese Weise können große XML Dokumente in handlichere Datenblöcke zerlegt werden.

Nach jeder Ausführung einer Pipeline-Stufe müssen die Ergebnisse übernommen werden. Da wir für die Repräsentation der Daten XML verwenden brauchen wir eine Möglichkeit, die Daten zu modifizieren, ohne sie vollständig in den Speicher zu laden oder jedes mal neu zu schreiben. Aus diesen Gründen wird auch hier wieder eine Indizierung durchgeführt. Diesmal allerdings vollständig, so dass eine modifizierender Zugriff auf alle Elemente möglich ist.

4.3 Ausführung

Die Analyse-Pipeline besteht aus einer Sequenz von verschiedenen Verarbeitungsstufen und jede von ihnen führt eine klar definierte Operation auf den Eingabedaten aus und produziert Ausgabedaten, die von den folgenden Stufen verwendet werden. Diese Tatsache macht eine parallele Ausführung aller Stufen vom Entwurf her unmöglich. Das Optimierungspotential liegt vielmehr in der parallelen Ausführung der einzelnen Stufen beziehungsweise Programmen [TXJ05] [Nat07]. Die Praxis hat jedoch gezeigt, dass die Input/Output (IO) Belastung der Dateiserver bei einer massiven parallelen Ausführung zu hoch ist. Auch ein Verteilen der Daten vor der Verarbeitung auf einen Knoten ist nicht praktikabel, da sie in der nächsten Stufe (und somit auf einem anderen Knoten) erneut gebraucht werden. Das hat zur Folge, dass jede Stufe alle Eingabedaten einmal lesen und die Ergebnisdaten zurückschreiben muss. Die beste Lösung ist eine unabhängige parallele Stapelverabeitung. Die Sequenzrohdaten werden in gleichgroße Mengen aufgeteilt und unabhängig voneinander verarbeitet. Auf diese Weise werden alle Intermediärformate lokal auf einem Knoten erzeugt. Dadurch reduziert sich der Zugriff auf den Dateiserver auf ein einmaliges Lesen der Eingabedaten und Zurückschreiben der Ergebnisse.

[4]NCBI Trace Archive - http://www.ncbi.nlm.nih.gov/Traces/trace.cgi?
[5]Tar - http://www.gnu.org/software/tar
[6]Standard Flowgram Format

5 Implementierung

Die Pipeline implementiert die gängigen Werkzeuge der Rohsequenzanalyse wie [EHW98] [CLW07] [ZSWM00]. Für die Ausführung steht ein Beowulf Cluster[7] zur Verfügung. Der Cluster wird durch eine SunGrid Engine[8] (SGE) gesteuert. Die SGE stellt die Werkzeuge und die Software Pakete für das Cluster Management und die verteilte Programmausführung bereit. Die Rohdaten befinden sich auf einem NFS[9] Dateiserver und sind von jedem Knoten des Clusters aus erreichbar. Der Workflow Interpreter wird vollständig in Perl entwickelt bis auf zwei externe Bibliotheken - Berkeley DB[10] zum Speichern eines Index und Expat[11] zum lesen von XML Dateien.

[7] Beowulf - http://beowulf.org
[8] SunGrid Engine - http://gridengine.sunsource.net
[9] Linux implementation of the Networking Filesystem - http://nfs.sourceforge.net
[10] Berkeley Database - http://www.oracle.com/technology/products/berkeley-db/db/index.html
[11] The Expat XML Parser - http://expat.sourceforge.net

Literatur

[ALV06] Adzhubei, Alexei A. ; Laerdahl, Jon K. ; Vlasova, Anna V.: preAssemble: a tool for automatic sequencer trace data processing. In: *BMC Bioinformatics* 7 (2006), S. 22. http://dx.doi.org/doi:10.1186/1471-2105-7-22. – DOI doi:10.1186/1471–2105–7–22

[CLW07] Chen, Yi-An ; Lin, Chang-Chun ; Wang, Chin-Di ; Wu, Huan-Bin ; Hwang, Pei-Ing: An optimized procedure greatly improves EST vector contamination removal. In: *BMC Genomics* 8 (2007). http://dx.doi.org/doi:10.1186/1471-2164-8-416. – DOI doi:10.1186/1471–2164–8–416

[DAC05] D'Agostino, Nunzio ; Aversano, Mario ; Chiusano, Maria L.: ParPEST: a pipeline for EST data analysis based on parallel computing. In: *BMC Bioinformatics* 6(Suppl 4) (2005), S. 9

[EHW98] Ewing, Brent ; Hillier, LaDeana ; Wendl, Michael C. ; ; Green, Phil: Base-Calling of Automated Sequencer Traces Using Phred. I. Accuracy Assessment. In: *Genome Research* 8 (1998), Nr. 3, S. 175–185

[GJB06] Goldberg, Susanne M. D. ; Johnson, Justin ; Busam, Dana ; Feldblyum, Tamara ; Ferriera, Steve ; Friedman, Robert ; Halpern, Aaron ; Khouri, Hoda ; Kravitz, Saul A. ; Lauro, Federico M. ; Li, Kelvin ; Rogers, Yu-Hui ; Strausberg, Robert ; Sutton, Granger ; Tallon, Luke ; Thomas, Torsten ; Venter, Eli ; Frazier, Marvin ; Venter, J. C.: A Sanger/pyrosequencing hybrid approach for the generation of high-quality draft assemblies of marine microbial genomes. In: *PNAS* 103 (2006), Nr. 30, S. 11240–11245

[Gor06] Gordon, David: Viewing and Editing Assembled Sequences Using Consed. In: Baxevanis, Andreas D. (Hrsg.): *Current Protocols in Bioinformatics*. John Wiley and Sons, 2006, Kapitel 11.2

[LAB06] Ludäscher, Bertram ; Altintas, Ilkay ; Berkley, Chad ; Higgins, Dan ; Jaeger, Efrat ; Jones, Matthew ; Lee, Edward A. ; Tao, Jing ; Zhao, Yang: Scientific workflow management and the Kepler system: Research Articles. In: *Concurr. Comput. : Pract. Exper.* 18 (2006), Nr. 10, S. 1039–1065. http://dx.doi.org/http://dx.doi.org/10.1002/cpe.v18:10. – DOI http://dx.doi.org/10.1002/cpe.v18:10. – ISSN 1532–0626

[LB05] Ludäscher, B. ; Bowers, S.: Actor-oriented design of scientific workflows. In: *Lecture Notes in Computer Science* x (2005), S. 369–384

[Mar08] Mardis, Elaine R.: The impact of next-genration sequencing technology on genetics. In: *Trends in Genetics* 24 (2008), S. 133–141

[Nat07] National Institutes of Health: *RepeatMasker on Biowulf*. Version: 2007. http://biowulf.nih.gov/apps/repeatmasker/index.html, Abruf: 12.12.2007

[OAF04] Oinn, T. ; Addis, M. ; Ferris, J. ; Marvin, D. ; ; Senger, M.: Taverna: a tool for the composition and enactment of bioinformatics workflows. In: *Bioinformatics* 20 (2004), Nr. 17, S. 3045–3054

[TXJ05] Tan, Guangming ; Xu, Lin ; Jiao, Yishan ; Feng, Shengzhong ; Bu, Dongbo ; Sun, Ninghui: An optimized algorithm of high spatial-temporal efficiency for Megablast. In: *Parallel and Distributed Systems, 2005. Proceedings. 11th International Conference on* Bd. 2, 2005, S. 703–708

[ZSWM00] Zhang, Zheng ; Schwartz, Scott ; Wagner, Lukas ; Miller, Webb: A greedy algorithm for aligning DNA sequences. In: *Journal of Computational Biology* 7 (2000), Nr. 1/2, S. 203–214