WHITE PAPER

The Impact of Large Shared Memory Computing Architectures in Genomics Workflows

Authors

Dr Haruna Cofer, Simon Appleby : Silicon Graphics International (SGI) : Bio Applications
Dr Mario Caccamo, Paul Fretter, Dr Timothy Stitt : The Genome Analysis Centre (TGAC) : Scientific Computing

Abstract

In this white paper that includes insight from leading genome facilities like The Genome Analysis Centre (TGAC), we will provide an analysis of why the SGI® UV™ architecture has become the platform of choice for many ambitious genome centers worldwide. The SGI UV architecture, now in its third generation, has arguably proven to be the most flexible, easy-to-use high-performance computing platform today. While traditional cluster nodes only provide 10's of processors and gigabytes (GB) of memory, a single SGI UV compute node provides shared access to 100's of processors and terabytes (TB) of memory. As a result, the SGI UV has seen success in complex pipeline implementation/integration, whole genome sequencing, rapid algorithm development, testing in data-realistic context environments, and responsive post-processing with large-scale data sets.

TGAC
The Genome Analysis Centre™

TABLE OF CONTENTS

1.0 Introduction to Genomic Workflows and Sequencing Centers

The genomics workflow is a collaborative effort among many research groups. Figure 1 illustrates a typical genomic workflow process from initial sampling to final analysis. A sample containing DNA/RNA is extracted from an organism (e.g. mouse or wheat) and isolated for laboratory sequencing in an intensive and specialised library preparation process. This preparation is then loaded onto the sequencing instrument and, depending on the platform and preparation, may be run for 5 to 10 days to generate 100's of GBs of data per day, requiring large amounts of storage. The raw data is then cleaned and filtered through a series of pre-processing steps to produce the best quality reads, reducing the data size in the range of 100's of MBs to 10's of GBs per day. These pre-processing operations are effectively simple string analysis problems that can typically run on a standard Linux cluster with relatively low memory requirements on the order of MBs to a few GBs of memory.

The sequence assembly that follows, however, is significantly more compute and memory intensive. The assembly process may take weeks to months on a low memory cluster node, while on single node SGI UV with 100's of GBs to TBs of memory, it may take only days to run. Further analysis and annotation (i.e. attempting to ascribe function to the DNA sequence information) of the assembled sequences can run on either a shared memory system like the SGI UV or a distributed memory system like the SGI® ICE™ platform, turning the data into valuable and practical insight. The genomics workflow can then continue with re-sampling and re-sequencing of different organisms and species.

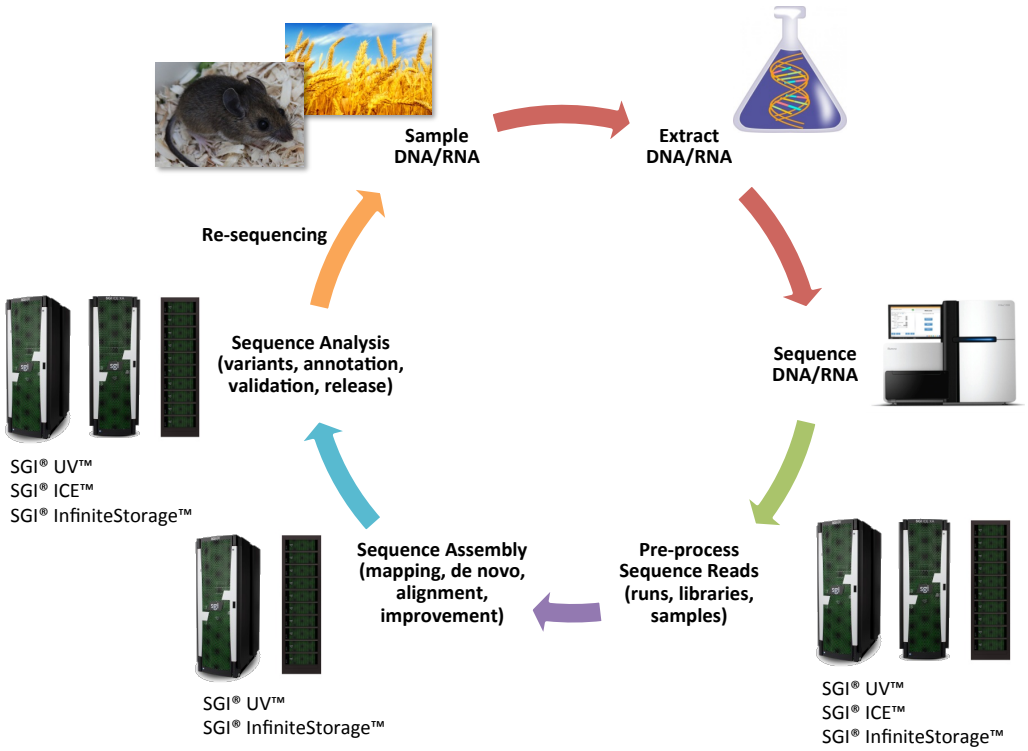


Figure 1

Interpreting these large amounts of genomic data into useful information and knowledge (Figure 2) presents a great challenge to computational infrastructures and their ability to process and store such vast amounts of complex data and information.

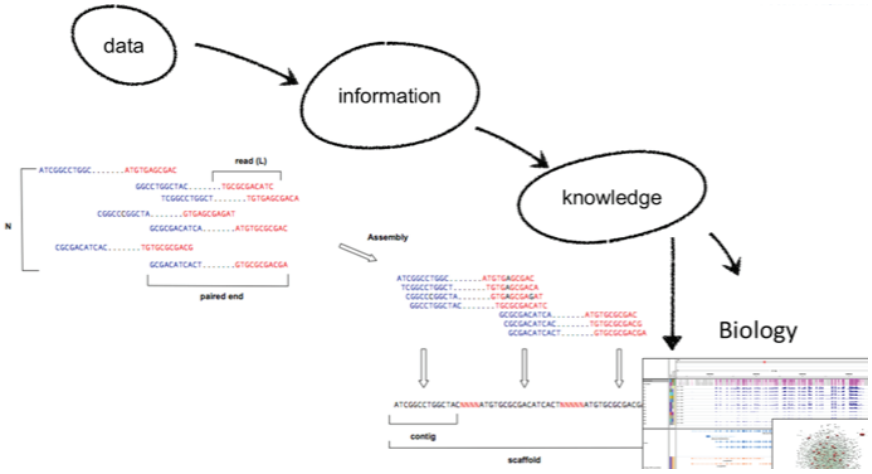


Figure 2

As sequencing technologies have become dramatically faster and more affordable, genome centers have accelerated the growth and development of state-of-the-art platforms for next-generation sequencing and computation (hard infrastructure), software algorithms and databases (soft infrastructure), a systems approach to genetic analysis, and the development of skills in new methodologies and approaches (training). All of these components provide the foundation for a "data-driven science" (Figure 3) that is able to sequence and interrogate entire genomes for a wide variety of species in a relatively short amount of time. As a result, significantly larger volumes of sequence data ranging from 100's of GBs to 10's of TBs in size are being produced, highlighting the need for a computing infrastructure with fast processing, large memory, rapid I/O and reliable storage.

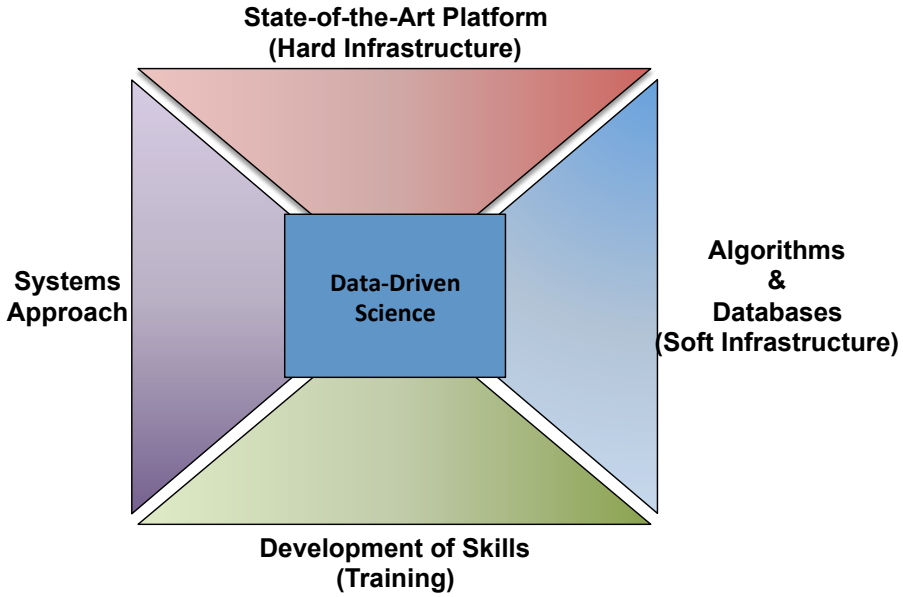


Figure 3

2.0 SGI UV for Sequence Assembly

Sequence assembly is the process of aligning and combining DNA or RNA fragments to assemble a final genome or transcriptome sequence. With the latest advances in next-generation sequencing (NGS) technologies, there are a growing number of sequence assembly algorithms for short-read fragments (50bp and longer). Unlike the overlap methods designed for the previous generation of long-read (up to 900bp) Sanger sequencers, these new assemblers must handle significantly larger quantities of data and require multiple parameter optimizations (i.e., repeated runs) in order to achieve a high level of genome coverage and quality in the final assembly. As a result, large shared memory computer systems like the SGI UV are critical for the successful assembly of large genomes and transcriptomes.

2.1 Methods for Short Read Assembly

There are two basic approaches used for short-read assemblies: 1) Mapping to an existing reference genome, and 2) *de novo* assembly without a reference genome. The mapping method attempts to overlay each read with the reference genome to obtain one or more alignments, and as such the time and memory requirement is typically linear with respect to read length. Some examples of popular mapping software include Bowtie2[1], BWA[2] and TopHat[3].

However, given that there are very few high-quality reference genomes available, most modern assemblers use a *de novo* method based on constructing a de Bruijn graph (Figure 4). The time and memory requirement for de Bruijn graph assembly is significantly greater than the mapping approach, but unlike *de novo* assemblies based on traditional pairwise alignments (Hamiltonian graphs, Figure 4c), de Bruijn graphs based on the Eulerian cycle (Figure 4d) are computationally more efficient and scalable to billions of graph nodes. For best performance, all of these graph nodes should be resident and quickly accessible from memory, further highlighting the need for a large shared memory system like the SGI UV.

A de Bruijn graph is a set of substrings of fixed length k (called k-mers) that contain all possible substrings in the data set exactly once. In de Bruijn graph assembly, each read from Figure 4b (5 reads from the circular genome in Figure 4a) is first broken into all possible sub-sequences of length k, or k-mers. A de Bruijn graph is then constructed (Figure 4d, k=3) that captures all of the overlaps of length k-1 between the k-mers. Each node represents a (k-1)-mer, and each edge between nodes represents a k-mer. A path is then calculated that traverses each edge just once, and this path or cycle represents an assembled genome sequence. The computational time to find such an Eulerian cycle is roughly proportional to the number of k-mers, making de Bruijn graph assemblies both scalable and tractable. As expected, the value of k significantly affects the quality of the final assembly and multiple assemblies may be performed to find the optimum k value, which depends on the coverage (how many reads overlap a given sub-sequence), read length, and error rate of the data set.

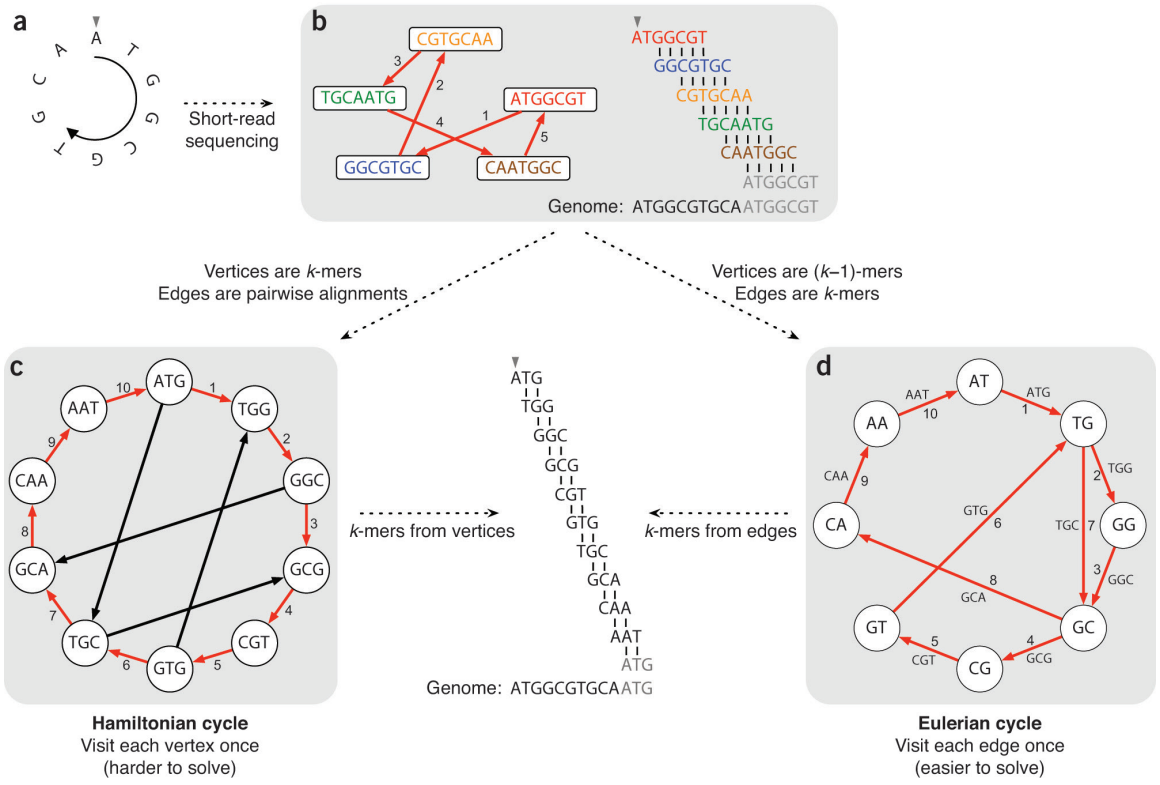


Figure 4: de Bruijn Graphs
(http://www.nature.com/nbt/journal/v29/n11/images/nbt.2023-F3.gif)[4]

In addition to sequencing a single genome, multiple genomes can be sequenced within a biological ecosystem or population. This research, called metagenomics, seeks to assemble the multiple genomes found in the different organisms of entire biological communities such as soil or the human gut. Metagenomics using similar de Bruijn graph analysis has the potential to generate far greater amounts of data than a single genome, and is considerably more complex due to the diversity of the sample. The elaborate nature of a metagenomic assembly can therefore require TBs of memory to model the multiple independent graph structures. From initial reads to the final assembly of sequences (contigs and scaffolds), being able to efficiently represent all of the subcomponents of the assembly in main memory is vital. As a result, large memory solutions like the SGI UV are extremely well placed to manage this requirement.

Finally, de Bruijn graph techniques are also used to detect and genotype genetic variants in an individual or population. Applications like Cortex[5] are not only able to assemble multiple eukaryotic genomes at once, but also detect both simple and complex structural variations, identify new sequences within population sequence data, and perform accurate variant calls – all without the need for a reference sequence. The generated de Bruijn graphs can be extremely large (on the order of TBs for large complex genomes), demanding similarly large amounts of memory, making the SGI UV architecture a necessity for future individualized analyses of the human genome.

2.2 Large Memory Requirements for Assembly

Some examples of popular *de novo* assembly tools include Velvet[6], SOAPdenovo[7], ABySS[8], ALLPATHS-LG[9], Ray[10], MIRA[11] and Trinity[12]. Such tools have been parallelized, but most of them run most efficiently on shared memory systems like the SGI UV. Velvet, SOAPdenovo, ALLPATHS-LG, MIRA and Trinity use OpenMP or pthreads, so they can only run on a single shared memory node, but only if there is enough memory on the node. Some applications such as Velvet will adjust their memory usage to accommodate a smaller memory configuration, but at the cost of significantly extra computing time. ABySS and Ray use MPI, so they can run on either a single shared memory or distributed-memory cluster system. The SGI UV proves very powerful for MPI based tools like ABySS and Ray because it can boost the performance of MPI applications through faster interconnects and MPI hardware acceleration via the Global Reference Unit (GRU). The SGI UV therefore offers outstanding flexibility for all sequence assembly applications, as it can easily and seamlessly run code that use the full suite of common parallel paradigms.

As mentioned previously, the memory requirements for *de novo* assembly increase dramatically with genome size, and with the correspondingly greater numbers of reads and errors within the data set. Assembling small genomes such as E. coli (4 million bases) and the larger rice genome (450 million bases) are fairly challenging but far from intractable, with memory requirements on the order of 10's of GB[13]. The human genome is approximately an order of magnitude larger (3 billion bases) and the wheat genome is even larger still (15-17 billion bases), with the assembly of the wheat genome requiring almost 2.5TB of memory using ALLPATHS-LG[14]. Large genomes may stress the memory available and how well the memory can be effectively and efficiently allocated, especially when memory is limited as on a cluster node. Even on a large shared memory node, one needs to configure a greater amount of memory than technically required, in order to ensure that the assembly will be able to allocate contiguous chunks of memory. Efficient memory allocators like Hoard[15] or TCMalloc[16] are useful in this regard, enabling larger assemblies for a given amount of memory.

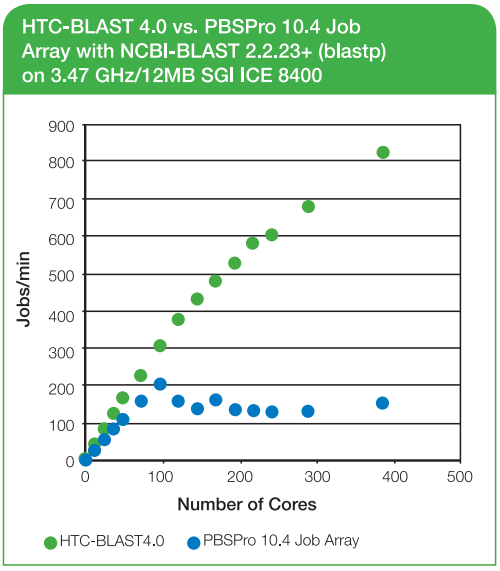
Another major issue faced by researchers of large scale *de novo* assembly is that one generally does not know a priori how much memory will be required to run a job to completion until the job simply terminates with an out-of-memory (OOM) error. The SGI UV platform, however, easily overcomes this obstacle by supporting up to 64TB of global shared memory in a single system image, thereby eliminating the memory limitation on current genomic workflows. Researchers from Oklahoma State University, for example, completed the largest metagenomics assembly to date by sequencing data from a soil metagenome that required 4TB of memory on Pittsburgh Supercomputing Center's Blacklight SGI UV 1000 system.

3.0 SGI UV for Sequence Analysis

After the sequence reads are successfully assembled, researchers can begin the task of analyzing the sequences for variants (single nucleotide polymorphisms, or SNPs, and insertions and deletions, or indels), structural and functional annotation, genome validation and final release. The SGI UV is an ideal platform for these subsequent analyses and especially as a speed search tool using algorithms such as BLAST[17] and HMMER[18], where newly assembled sequences are compared against large GB to TB sized databases of known DNA and protein sequences. When these database files are used by an application, they are automatically placed in the filesystem buffer cache and remain there for other analysis jobs to use until the memory is needed by another process. Once the database files are in SGI UV memory, they can be accessed in under a microsecond with NUMAlink shared memory – randomly reading anywhere. Empirically, in-memory data on SGI UV is 1,000x faster than disk (HDD) and 100x faster than flash disk (SSD). Loading entire database files into memory is therefore crucial to the compute performance of the post-assembly analysis, annotation and validation of genomic sequences.

3.1 SGI High Throughput Computing (HTC) for Sequence Analysis

In addition to a large shared memory capable of holding terabyte-scale databases, SGI applications engineers have developed a wrapper program, the HTC[19] (High Throughput Computing) Wrapper Program for Bioinformatics, that dramatically improves the throughput performance of sequence analysis jobs on an SGI system such as the SGI UV. Standalone algorithms like BLAST and HMMER are extremely fast but limited in their scalability, which can impede overall performance and efficient usage of the system. Also, the start-up overhead associated with launching jobs can become very significant when multiplied by thousands to hundreds of thousands of jobs. Freely distributed by SGI as a single binary download, SGI's HTC transparently reads all the inputs, load balances the jobs, and then submits them to maximize system utilization and efficiency. HTC has been shown to demonstrate significantly higher performance and scalability than a general batch scheduler for processing thousands of query sequences, accelerating the throughput of any user-supplied script as well as a variety of bioinformatics applications already available such as BLAST, FASTA[20], ClustalW[21], HMMER and Wise2[22].



Queries: 10,000 sequences from RefSeq protein database (3,221,557 residues in 10,000 sequences)

Database: Non-Redundant Protein Database (1,336,756,300 residues in 3,878,509 sequences)

Overall, the SGI UV system provides a maximum dynamic range in computational capacity which is critical for leading genomic research. There are currently no alternative systems that can handle the large scale computational integration of TB sized data structures needed in the support of genomic workflows and next-generation sequence assembly.

4.0 Additional SGI UV Features for Genomics Research

4.1 Enabling Checkpoint Restart

Unpredictable runtime is a major issue faced by researchers who must run their assemblies on a system shared with other users. For a variety of reasons, such as a batch environment or maintenance downtime, sequence assembly jobs that have been running for days to weeks can unexpectedly terminate. One solution to this problem is using application checkpointing software such as DMTCP[23] which provides a lightweight user level tool to checkpoint and restart sequence assembly jobs. Because these jobs can be TBs in size, the system must have enough extra memory to support the caching of TB sized checkpoint files. The SGI UV with its large memory capacity can not only alleviate concerns about memory limitations but also time limitations, thereby providing a powerful computational resource for all genomics workflows.

4.2 Optimizing Storage and I/O

The SGI UV system supports direct attached storage, which makes large genomic datasets immediately and transparently accessible to any processor on the system. Standard cluster systems employ traditional parallel file systems that are only accessible over a network, which is significantly slower because of the remote latency. However, the SGI UV does not suffer from this I/O bottleneck and file performance is further enhanced by its XFS file system compared to standard ext4. Furthermore, SGI's optimized I/O libraries such as Flexible File I/O (FFIO) can significantly improve I/O performance, in particular on next-generation sequence utilities like the Broad Institute's Picard[24] tool set.

4.3 Accelerating Computations

Finally, the SGI UV can be further enhanced with coprocessors such as Intel® Xeon® Phi™ and GPU accelerators such as NVIDIA® Quadro® and NVIDIA® Tesla®. With the growing number of accelerated applications for genomics, the SGI UV is able to provide even faster computations using these advanced processing architectures.

5.0 Customer Case Study: The Genome Analysis Centre (TGAC)

The Genome Analysis Centre (TGAC) based in Norwich, UK, specializes in genomics and bioinformatics with a focus on analysis and interpretation of plant, animal and microbial genomes. Launched in July 2009, TGAC has grown its teams to over 70 members, with half of the institute focused on the interpretation, assembly and analysis of datasets generated by in-house sequencers. TGAC has installed three large SGI UV systems based on Intel x86-64 processors, the most recent being an SGI UV 2000 with 20TB of globally shared memory and an SGI optimized shared file system. Dr Mario Caccamo, Director of TGAC, states "the main benefit of using such a system is the ability to assemble and analyse large and complex genome sequences in memory. We were experiencing memory limitations in standard cluster x86 hardware and thus had difficulty in assembling large and complex genomes."

5.1 Current Computational Needs

As described previously, sequence assemblies by themselves are extremely memory intensive and further complicated by the unpredictability of the expected memory usage and run time. Running such assemblies on a standard cluster may require multiple trial and error attempts, as well as creative approaches to job management. TGAC therefore decided that only large cache-coherent shared memory systems such as the SGI UV could complete their assemblies efficiently and effectively, while still using a standard x86-64 Linux based platform with generic open source support. Access to a large memory system furthermore enables their researchers to run community and commodity bioinformatics software with ease and transparency, making this capability a critical and fundamental requirement. Examples of key large shared and distributed (MPI) memory applications used at TGAC include ABySS, Velvet and their novel sequence assembly tool, Cortex.

Significant efforts are being made both in-house and within the greater bioinformatics community to re-engineer algorithms for efficient parallel execution. Parallel programming paradigms for shared memory systems such as OpenMP and pthreads are generally the easiest and best performing methods, but distributed programming methods like MPI are also growing in implementation and popularity. To support both types of parallelism, TGAC recognized the need to reduce latency and improve bandwidth of memory and communication between process threads beyond what was achievable with their current cluster system. The SGI UV with its low latency and fast NUMAlink interconnect was therefore TGAC's platform of choice for achieving reduced execution times of their parallel genomic workflow software.

Most of the bioinformatics tools used at TGAC perform file-based I/O, with individual file sizes varying from a few megabytes to a few gigabytes. However, the typical dataset being processed consists of multiple files with aggregate file sizes ranging from tens to hundreds of gigabytes. Examples of analysis tools used at TGAC include BioScope[25], BLAST, EMBOSS, CASAVA[26], MEGAN[27], KAT[28], Amos[29], BioPerl[30], and Pacific Bioscience's SMRTPortal[31] software for PacBio RS sequencing data analysis. Some applications require access to local database servers, and other software such as Mira and InterProScan[32] require fast access to a local file system (e.g., /tmp). Pipeline processing elements can also be memory intensive and may require more than 400GB of memory in a single space. Because of the local I/O requirements and the expectation that memory usage would increase significantly in the near future, TGAC decided to purchase a minimum of 6TB of globally shared memory with locally (direct) attached storage. TGAC's UV ecosystem now includes two UV 100 systems, each with 768 cores and 6TB RAM, plus a UV 2000 with 2,560 cores and 20TB RAM. The UV 2000 also includes 32 Intel® Xeon Phi™ processors to allow computational staff to evaluate the processor as a potential offload platform for bioinformatics codes.

5.2 Preparing for the Future

While the current production processes on TGAC's cluster system are reasonably well understood and defined, the future demands for downstream processing and analysis are more variable and less determined, mainly shaped by the rapidly changing needs of current and future customers and scientific collaborations. The processing and resource requirements can vary significantly from project to project, making it of utmost importance that the supporting computational infrastructure be easily adaptable and fully expandable to meet rapidly evolving scientific needs. TGAC's cutting edge algorithmic development keeps pace with these growing demands and significant attention is paid to reducing the memory footprint, but key programmatic elements must still be handled in a single large memory space. In conclusion, the standard Linux-based SGI UV with its industry leading support for large shared memory and processor configurations is perfectly positioned to handle the extremes of computational productivity and storage for current and future genomic workflows.

6.0 Summary of SGI UV Impact on Genomics Workflows

The SGI UV platform provides reliability, efficiency, and leading performance for genomics applications ranging from in-memory assemblies and databases, to diverse sets of data and compute-intensive analyses, while maintaining a familiar programming and administration environment via the standard Linux® operating system. The total cost of ownership (TCO) is inherently lower due to its single-system administration. The researcher's workflow and overall time to solution is accelerated by running all of the pre-processing, assembly and analysis operations on one low TCO system using SGI enhanced hardware and software solutions, without the need to move data.

SGI has been a leader in the life sciences community for more than 20 years delivering computational solutions for discovery research organizations in pharmaceutical, chemical and biotechnology as well as in academic and national labs. This background, along with industry expertise and a working knowledge of customer applications and issues, positions SGI to make a real difference in delivering genomics solutions that are not only proven and productive, but also revolutionary in their ability to drive discovery and reduce time to insight. Today, life sciences' dependence on HPC is greater than ever before. In the genomics world, where data sets are massive and computational challenges are formidable, the SGI UV platform offers significant advantages to speed and accelerate discovery.

7.0 References

1. Langmead, B., and S. Salzberg. "Fast gapped-read alignment with Bowtie2." Nature Methods 9: 357-59. Print.
2. Li, H., and R. Durbin. "Fast and accurate long-read alignment with Burrows-Wheeler Transform." Bioinformatics (2010): n. pag. Print.
3. Trapnell, C., L. Pachter, and S.L. Salzberg. "TopHat: discovering splice junctions with RNA-Seq." Bioinformatics 25.9 (2009): 1105-11. Print.
4. Compeau, Phillip E., Pavel A. Pevzner, and Glen Tesler. "How to apply de Bruijn graphs to genome assembly." Nature Biotechnology 29.11 (2011): 987-91. Print.
5. Iqbal, Zamin, et al. "De novo assembly and genotyping of variants using colored de Bruijn graphs." Nature Genetics 44 (2012): 226-32. Print.
6. Zerbino, D.R., and E. Birney. "Velvet: algorithms for de novo short read assembly using de Bruijn graphs." Genome Research 18: 821-829. Print.
7. Luo, et al. "SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler." GigaScience 1 (2012): 18. Print.

8. Simpson J.T., et al. “ABySS: A parallel assembler for short read sequence data.” Genome Research (2009): n. pag. Print.

9. Gnerre, S., et al. “High-quality draft assemblies of mammalian genomes from massively parallel sequence data.” Proceedings of the National Academy of Sciences USA 108.4 (2011): 1513-1518. Print.

10. Boisvert, Sébastien, et al. “Ray Meta: scalable de novo metagenome assembly and profiling.” Genome Biology (2012): n. pag. Print.

11. Chevreux, Bastien, et al. “Genome Sequence Assembly Using Trace Signals and Additional Sequence Information.” Computer Science and Biology: Proceedings of the German Conference on Bioinformatics 99 (1999): 45-56. Print.

12. Grabherr, M.G., et al. “Full-length transcriptome assembly from RNA-seq data without a reference genome.” Nature Biotechnology 29.7 (2011): 644-52. Print.

13. Ye, Chengxi, et al. “Exploiting sparseness in de novo genome assembly.” BMC Bioinformatics 13 S6 (2012): S1. Print.

14. Blood, P.D., S. Marcus, and M.C. Schatz. “Largescale sequencing and assembly of cereal genomes using Blacklight.” XSEDE ‘14 Proceedings of the 2014 Annual Conference on Extreme Science and Engineering Discovery Environment. New York: ACM, 2014. N. pag. Print.

15. Berger, E. D., et al. “Hoard: A Scalable Memory Allocator for Multithreaded Applications” ASPLOS-IX. Proceedings of the ninth international conference on Architectural support for programming languages and operating systems. New York: ACM, 2000. 117–128. Print.

16. “gperftools - Fast, multi-threaded malloc() and nifty performance analysis tools.” Google Code. Google, n.d. Web. 10 June 2015. <https:// code.google.com/p/gperftools/>.

17. Altschul, S., W. Gish, et al. “Basic local alignment search tool.” Journal of Molecular Biology 215.3 (1990): 403–410. Print.

18. Finn, R.D., J. Clements, and S.R. Eddy. “HMMER web server: interactive sequence similarity searching.” Nucleic Acids Research 39 (2011): W29-W37.

19. Cofer, H. SGI® High Throughput Computing (HTC) Wrapper Program for Bioinformatics on SGI ICE™ and SGI UV™ Systems. N.p.: Silicon Graphics International, 2012. Print.

20. Pearson, W.R., and D.J. Lipman. “Improved tools for biological sequence comparison.” Proceedings of the National Academy of Sciences USA. 85.5 (1988): 2444-2448. Print.

21. Larkin, M.A., et al. “Clustal W and Clustal X version 2.0.” Bioinformatics 23 (2007): 2947-2948. Print.

22. Birney, E. “Wise2.” EMBL European Bioinformatics Institute. EMBL-EBI, n.d. Web. 10 June 2015. <https://www.ebi.ac.uk/~birney/wise2/>.

23. Ansel, Jason, Kapil Arya, and Gene Cooperman. “DMTCP: Transparent Checkpointing for Cluster Computations and the Desktop.” 23rd IEEE International Parallel and Distributed Processing Symposium (IPDPS’09). Rome, Italy. N.p.: IEEE, 2009. N. pag. Print.

24. “Picard Tools - By Broad Institute.” Broad Institute, n.d. Web. 10 June 2015. <http://broadinstitute.github.io/picard>.

25. SOLiD™ BioScope™ Software. N.p.: Life Technologies Corporation, 2010. Print.

26. “CASAVA Support.” Illumina Support. Illumina, Inc., n.d. Web. 10 June 2015. <http://support.illumina.com/sequencing/sequencing_software/casava.html>.

27. “MEGAN5 - MEtaGenome ANalyzer — Algorithms in Bioinformatics.” Universität Tübingen, n.d. Web. 10 June 2015. <http://www-ab.informatik.uni-tuebingen.de/software/megan5>.

28. “kat download | SourceForge.net.” SourceForge - Download, Develop and Publish Free Open Source Software. Slashdot Media, n.d. Web. 10 June 2015. <http://sourceforge.net/projects/kat/>.

29. “AMOS - Statistics Solutions.” Statistics Solutions: Dissertation and Research Consulting For Statistical Analysis. Statistics Solutions, n.d. Web. 10 June 2015. <http://www.statisticssolutions.com/amos/>.

30. “BioPerl.” BioPerl. N.p., n.d. Web. 10 June 2015. <http://www.bioperl.org/>.

31. “Pacific Biosciences: Analysis.” Pacific Biosciences. Pacific Biosciences of California, Inc. , n.d. Web. 10 June 2015. <http://www.pacificbiosciences.com/products/software/secondary-analysis/>.

32. Jones, P., et al. “InterProScan 5: genome-scale protein function classification.” Bioinformatics (2014): n. pag. Print.

8.0 About SGI

SGI is a global leader in high performance solutions for compute, data analytics and data management that enable customers to accelerate time to discovery, innovation, and profitability. For more information about the UV product line, please visit www.sgi.com/uv.

To see how SGI can help you, please contact: Simon Appleby – Life Sciences Manager – EMEA: sappleby@sgi.com or James Reaney - Senior Director, Research Markets; reaney@sgi.com

Global Sales and Support: sgi.com