

Report from Dagstuhl Seminar 16351

Next Generation Sequencing – Algorithms, and Software For Biomedical Applications

Edited by

Gene Myers¹, Mihai Pop², Knut Reinert³, and Tandy Warnow⁴

1 MPI – Dresden, DE, myers@mpi-cbg.de

2 University of Maryland – College Park, US, mpop@umd.edu

3 FU Berlin, DE, knut.reinert@fu-berlin.de

4 University of Illinois – Urbana-Champaign, US, warnow@illinois.edu

Abstract

Next Generation Sequencing (NGS) data have begun to appear in many applications that are clinically relevant, such as resequencing of cancer patients, disease-gene discovery and diagnostics for rare diseases, microbiome analyses, and gene expression profiling. The analysis of sequencing data is demanding because of the enormous data volume and the need for fast turnaround time, accuracy, reproducibility, and data security. This Dagstuhl Seminar aimed at a free and deep exchange of ideas and needs between the communities of algorithmicists and theoreticians and practitioners from the biomedical field. It identified several relevant fields such as data structures and algorithms for large data sets, hardware acceleration, new problems in the upcoming age of genomes, etc., which were discussed in breakout groups.

Seminar August 28 to September 2, 2016 – <http://www.dagstuhl.de/16351>

1998 ACM Subject Classification D.2.11 Software Architectures, D.2.13 Reusable Software, D.2.2 Design Tools and Techniques, E.1 Data Structures, J.3 Life and Medical Sciences

Keywords and phrases Cancer, DNA Sequence Assembly, Expression Profiles, Next Generation Sequencing, Sequence analysis, Software Engineering (Tools & Libraries)

Digital Object Identifier 10.4230/DagRep.6.8.91

Edited in cooperation with German Tischler

1 Executive summary

Gene Myers

Mihai Pop

Knut Reinert

Tandy Warnow

License © Creative Commons BY 3.0 Unported license
© Gene Myers, Mihai Pop, Knut Reinert, and Tandy Warnow

Motivation

In recent years, Next Generation Sequencing (NGS) data have begun to appear in many applications that are clinically relevant, such as resequencing of cancer patients, disease-gene discovery and diagnostics for rare diseases, microbiome analyses, and gene expression profiling, to name but a few. Other fields of biological research, such as phylogenomics, functional genomics, and metagenomics, are also making increasing use of the new sequencing technologies.



Except where otherwise noted, content of this report is licensed under a Creative Commons BY 3.0 Unported license

Next Generation Sequencing – Algorithms, and Software For Biomedical Applications, *Dagstuhl Reports*, Vol. 6, Issue 8, pp. 91–130

Editors: Gene Myers, Mihai Pop, Knut Reinert, and Tandy Warnow



DAGSTUHL
REPORTS Dagstuhl Reports

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The analysis of sequencing data is demanding because of the enormous data volume and the need for fast turnaround time, accuracy, reproducibility, and data security. Addressing these issues requires expertise in a large variety of areas: algorithm design, high performance computing on big data (and hardware acceleration), statistical modeling and estimation, and specific domain knowledge for each medical problem. In this Dagstuhl Seminar we aimed at bringing together leading experts from both sides – computer scientists including theoreticians, algorithmicists and tool developers, as well as leading researchers who work primarily on the application side in the biomedical sector – to discuss the state-of-the art and to identify areas of research that might benefit from a joint effort of all the groups involved.

Goal of the seminar

The key goal of this seminar was a free and deep exchange of ideas and needs between the communities of algorithmicists and theoreticians and practitioners from the biomedical field. This exchange should have triggered discussions about the implications that new types of data or experimental protocols have on the needed algorithms or data structures.

Results

We started the seminar with a number of *challenge talks* to encourage discussion about the various topics introduced in the proposal. Before the seminar started we identified three areas the participants were most interested in, namely:

1. Data structures and algorithms for large data sets, hardware acceleration
2. New problems in the upcoming age of genomes
3. Challenges arising from new experimental frontiers and validation

For the first area Laurent Mouchard, Gene Myers, and Simon Gog presented results and challenges; for the second area Siavash Mirarab, Niko Beerenwinkel, Shibu Yooseph, and Kay Nieselt introduced some thoughts; and finally, for the last area, Jason Chin, Ewan Birney, Alice McHardy, and Pascal Costanza talked about challenges. For most of those talks the abstracts can be found below. Following this introductory phase, the participants organized themselves into various working groups the topics of which were relatively broad. Those first breakout groups were about

- Haplotype phasing
- Big data
- Pangenomics data representation
- Cancer genomics
- Metagenomics
- Assembly

The results of the groups were discussed in plenary sessions interleaved with some impromptu talks. As a result the participants split up into smaller, more focused breakout groups that were received very well. Indeed, some participants did already extend data formats for assembly or improved recent results on full text string indices.

Based on the initial feedback from the participants we think that the topic of the seminar was interesting and led to a lively exchange of ideas. We thus intend to revisit the field in the coming years in a Dagstuhl seminar again, most likely organized by different leaders of the field in order to account for these upcoming changes. In such a seminar we intend to encourage more people from clinical bioinformatics to join into the discussions.

2 Table of Contents

Executive summary

<i>Gene Myers, Mihai Pop, Knut Reinert, and Tandy Warnow</i>	91
--	----

Overview of Talks

Computational challenges in cancer genomics <i>Niko Beerenwinkel</i>	96
New advances in Sequencing Technology <i>Ewan Birney</i>	96
The art and science that we can learn from assembly graphs <i>Jason Chin</i>	96
Non-algorithmic aspects of sequencing software <i>Pascal Costanza</i>	97
Challenges in designing a library of practical compact data structures <i>Simon Gog</i>	98
Gene-Centric Assembly <i>Daniel H. Huson</i>	98
Computational Pan-Genomics: Status, Promises and Challenges <i>Tobias Marschall</i>	98
Challenges in organizing a metagenomic benchmarking challenge <i>Alice Carolyn McHardy</i>	99
Upcoming challenges in phylogenomics <i>Siavash Mirarab</i>	99
Recent advances and future challenges in BWT <i>Laurent Mouchard</i>	100
Examples where theory fails in practice and practice needs some theory <i>Gene Myers</i>	100
Pangenome Variant Calling <i>Veli Mäkinen</i>	101
Challenges of ancient genomics and pan-genomics <i>Kay Nieselt</i>	101
Data structures to employ embeddings of strings under edit distances to vectors under Hamming distance <i>S. Cenk Sahinalp</i>	102
Ensembles of HMMs <i>Tandy Warnow</i>	102
Three problems in metagenomics <i>Shibu Yooseph</i>	103

Working groups

Single-cell cancer genomics: variant calling & phylogeny <i>Niko Beerenwinkel, Mohammed El-Kebir, Gunnar W. Klau, Tobias Marschall, and S. Cenk Sahinalp</i>	104
Cancer genomics <i>Mohammed El-Kebir, Niko Beerenwinkel, Christina Boucher, Anne-Katrin Emde, Birte Kehr, Gunnar W. Klau, Pietro Lio', Siavash Mirarab, Luay Nakhleh, Esko Ukkonen, and Tandy Warnow</i>	106
Software libraries for indexing <i>Simon Gog, Pascal Costanza, Anthony J. Cox, Fabio Cunial, Hannes Hauswedell, André Kahles, Ben Langmead, Laurent Mouchard, Gene Myers, Enno Ohlebusch, Simon J. Puglisi, Gunnar Rätsch, Knut Reinert, Bernhard Renard, Enrico Siragusa, German Tischler, and David Weese</i>	109
Assembly <i>Gene Myers, Jason Chin, Richard Durbin, Mohammed El-Kebir, Anne-Katrin Emde, Birte Kehr, Oliver Kohlbacher, Veli Mäkinen, Alice Carolyn McHardy, Laurent Mouchard, Kay Nieselt, Adam M. Phillippy, Tobias Rausch, Peter F. Stadler, Granger Sutton, German Tischler, and David Weese</i>	112
Big data <i>Gene Myers, Ewan Birney, Pascal Costanza, Anthony J. Cox, Fabio Cunial, Richard Durbin, Simon Gog, Hannes Hauswedell, Birte Kehr, Ben Langmead, Laurent Mouchard, Enno Ohlebusch, Adam M. Phillippy, Mihai Pop, Simon J. Puglisi, Tobias Rausch, Karin Remington, S. Cenk Sahinalp, Peter F. Stadler, and German Tischler</i>	114
Structural Variant Detection <i>Gene Myers, Jason Chin, Mohammed El-Kebir, Anne-Katrin Emde, Birte Kehr, Veli Mäkinen, Tobias Marschall, Adam M. Phillippy, Mihai Pop, Karin Remington, S. Cenk Sahinalp, and Granger Sutton</i>	116
Visualization Group <i>Gene Myers, Jason Chin, Mohammed El-Kebir, Anne-Katrin Emde, Birte Kehr, Veli Mäkinen, Tobias Marschall, Adam M. Phillippy, Karin Remington, S. Cenk Sahinalp, and Granger Sutton</i>	119
Metagenomics <i>Mihai Pop, Pascal Costanza, Anthony J. Cox, Fabio Cunial, Simon Gog, Hannes Hauswedell, Daniel H. Huson, André Kahles, Pietro Lio', Alice Carolyn McHardy, Siavash Mirarab, Kay Nieselt, Enno Ohlebusch, Simon J. Puglisi, Gunnar Rätsch, Karin Remington, Bernhard Renard, Enrico Siragusa, Tandy Warnow, and Shibu Yooseph</i>	120
Haplotype Phasing <i>Knut Reinert, Niko Beerenwinkel, Jason Chin, Richard Durbin, Mohammed El-Kebir, Anne-Katrin Emde, Gunnar W. Klau, Veli Mäkinen, Tobias Marschall, Alice Carolyn McHardy, Siavash Mirarab, Kay Nieselt, Bernhard Renard, Enrico Siragusa, Peter F. Stadler, Granger Sutton, Tandy Warnow, and David Weese</i>	125

Pan-Genomics
Knut Reinert, Jason Chin, Fabio Cunial, Simon Gog, André Kahles, Birte Kehr, Oliver Kohlbacher, Ben Langmead, Alice Carolyn McHardy, Siavash Mirarab, Kay Nieselt, Enno Ohlebusch, Adam M. Phillippy, Simon J. Puglisi, Gunnar Rättsch, Karin Remington, Bernhard Renard, Peter F. Stadler, Granger Sutton, German Tischler, and Shibu Yooseph 126

Participants 130

3 Overview of Talks

3.1 Computational challenges in cancer genomics

Niko Beerenwinkel (ETH Zürich – Basel, CH)

License  Creative Commons BY 3.0 Unported license
© Niko Beerenwinkel

Cancer genomics has seen tremendous advancements with the arrival of cost-effective high-throughput sequencing. These technologies allow for analyzing cancer samples in unprecedented detail. At the same time, the resulting sequencing data poses a range of new computational challenges in analyzing and interpreting the data. These challenges include (1) read mapping and mutation calling in mixed tumor samples, including low-frequency variants; (2) detection of complex genomic alterations, which are common in cancer genomes; (3) inferring the clonal structure of mixed tumor samples from bulk sequencing data; (4) reconstructing the evolutionary history of a tumor, i.e., solving the tumor phylogeny problem; (5) reconstructing tumor phylogenies from single-cell sequencing data, and (6) predicting cancer evolution by learning models from independent observations across tumor samples from different patients. Approaches to all of these challenges exist, but most are inherently difficult or even mathematically ill-posed. Progress with these challenges is likely to have an impact on cancer diagnostics and treatment.

3.2 New advances in Sequencing Technology

Ewan Birney (European Bioinformatics Institute – Cambridge, GB)

License  Creative Commons BY 3.0 Unported license
© Ewan Birney

I will present an overview of the features of new sequencing technology, in particular PacBio and Oxford Nanopore. Both produce long reads with somewhat higher error rates than Illumina short read sequencing. The error rate though is manageable as has been shown in particular with PacBio data. Both systems work asynchronously with individual reads being produced. In the case of Oxford Nanopore, the sequencing process can be stopped early in a read and a new read resampled in real-time. This provides new avenues of algorithms which combine decision making in real time with sampling management.

Note: I am a paid consultant to Oxford Nanopore, and thus I am very explicit about this conflict of interest

3.3 The art and science that we can learn from assembly graphs

Jason Chin (Pacific Biosciences – Menlo Park, US)

License  Creative Commons BY 3.0 Unported license
© Jason Chin

In an overlap-layout-consensus assembler, the assembly graph constructed from read overlaps is the major data structure for generating contigs. Repeat-induced ambiguities within the graph are typically removed by analyzing local neighboring subgraph, defined as a subgraph

of a selected node or an edges and its nearest neighbors, properties. Contigs are constructed after removing those ambiguities. However, the heuristic rules to remove the ambiguities may also remove useful information that can be used for improving genome assembly and understand local genome structure.

By analyzing non-local graph structures (e.g. the subgraph within certain distance from a vertex), we can recover such missing information and reveal important biological information within the data. For example, heterozygous variants between haplotypes within a diploid genome usually create “bubbles” in the assembly graph. Identifying and analyzing such bubbles can lead to a full haplotype resolved assembly. Local unresolved repeats also created local tangled sub-graphs which might break contigs. In such case, if we can still identify unique source and sink of the subgraph, we can generate the linkage information to connect contigs into as “extend contigs”. While some ambiguities remain in the tangled region, the extend contigs will contain all sequence information of the repeat regions.

Here are some related challenges for utilizing the assembly graph to extract more biological information:

1. Utilize the assembly graph information to define the “quality value” indicating uncertainties or errors at a given point of the contigs.
2. Understand whether there are systematic patterns of local repeats.
3. Develop algorithms for combining different data types at assembly graph level for scaffolding and resolving ambiguities.

3.4 Non-algorithmic aspects of sequencing software

Pascal Costanza (Intel Corporation, BE)

License  Creative Commons BY 3.0 Unported license
© Pascal Costanza

When composing tools to create sequencing pipelines, the most widely used approach to pass intermediate results from one tool to the next is through intermediate files. This limits the scalability of such pipelines when trying to take advantage of multiple cores due to Amdahl’s Law, since the file transfer from one tool to the next is a sequential bottleneck. We have shown in previous work that this limitation can be overcome by grouping several steps in a pipeline into a single tool and keeping all data in memory. Upcoming new memory technologies will make it more and more feasible to keep large amounts of data in memory - however, there is currently no good solution for allowing several tools written in different programming languages to equally take advantage of such in-memory representations of sequencing data and allow them to collaborate without going through the bottleneck of file transfers. Memory-mapped binary file formats that can be accessed through shared memory may be an answer, but there are open challenges that need to be addressed to make this practical.

3.5 Challenges in designing a library of practical compact data structures

Simon Gog (KIT – Karlsruher Institut für Technologie, DE)

License © Creative Commons BY 3.0 Unported license
© Simon Gog

In this talk we discuss the challenges in designing and maintaining a data structure library which enables researchers in Bioinformatic to build tools which can handle large datasets. Three of the main challenges in the moment is to improve the construction process of index structures, to identify primitives which allow the composition of structures which can deal with highly-repetitive data, and to add support for dynamic operations like inserting and deleting sequences.

We exemplify the impact of an improvement of a basic data structure to applications by the use of the partitioned Elias-Fano (PEF) encoding in Compressed Suffix Arrays. PEF was developed in the Information Retrieval field but we think that it has also impact on Bioinformatic applications.

We provide a tutorial for participant interested in space-efficient data structures here: <https://github.com/simongog/sigir16-topkcomplete>

3.6 Gene-Centric Assembly

Daniel H. Huson (Universität Tübingen, DE)

License © Creative Commons BY 3.0 Unported license
© Daniel H. Huson

Assembly of microbiome sequencing datasets is generally a difficult problem in practice. Some questions require the genes to be assembled, rather than genomes. Gene-centric assembly aims at assembling all reads that are recruited to a specific gene family. A first simple approach is to use BLASTX or DIAMOND (or an HMM-based approach) to align reads to references representing a given gene family and then to pass the reads to a full-featured assembler such as IDBA. We described so-called protein-reference guided assembly that aims at using protein alignments to detect DNA overlaps between reads recruited to a given gene family. Such an approach is implemented in the MEGAN software and this was briefly demonstrated.

3.7 Computational Pan-Genomics: Status, Promises and Challenges

Tobias Marschall (Universität des Saarlandes, DE)

License © Creative Commons BY 3.0 Unported license
© Tobias Marschall

Main reference The Computational Pan-Genomics Consortium, “Computational Pan-Genomics: Status, Promises and Challenges”, Briefings in Bioinformatics, pp. 1–18, 2016.

URL <http://dx.doi.org/10.1093/bib/bbw089>

Many disciplines, from human genetics and oncology to plant breeding, microbiology and virology, commonly face the challenge of analyzing rapidly increasing numbers of genomes. In case of Homo sapiens, the number of sequenced genomes will approach hundreds of thousands

in the next few years. Simply scaling up established bioinformatics pipelines will not be sufficient for leveraging the full potential of such rich genomic datasets. Instead, novel, qualitatively different computational methods and paradigms are needed. We will witness the rapid extension of computational pan-genomics, a new sub-area of research in computational biology. In this paper, we generalize existing definitions and understand a pan-genome as any collection of genomic sequences to be analyzed jointly or to be used as a reference. We examine already available approaches to construct and use pan-genomes, discuss the potential benefits of future technologies and methodologies, and review open challenges from the vantage point of the above-mentioned biological disciplines. As a prominent example for a computational paradigm shift, we particularly highlight the transition from the representation of reference genomes as strings to representations as graphs. We outline how this and other challenges from different application domains translate into common computational problems, point out relevant bioinformatics techniques and identify open problems in computer science. With this review, we aim to increase awareness that a joint approach to computational pan-genomics can help address many of the problems currently faced in various domains. (Abstract taken from DOI: 10.1093/bib/bbw089, CC-BY 3.0)

3.8 Challenges in organizing a metagenomic benchmarking challenge


Alice Carolyn McHardy (Helmholtz Zentrum – Braunschweig, DE)

License  Creative Commons BY 3.0 Unported license
© Alice Carolyn McHardy

The computational analysis of metagenomic NGS data sets is a rapidly evolving field. The Initiative for the Critical Assessment of Metagenome Interpretation (CAMI) aims to evaluate methods in metagenomics independently, comprehensively and without bias. The first CAMI challenge has been run in 2015. We find that the most important challenges of organizing such a challenge are to (i) engage both the method developer and the applied metagenomics fields, (ii) to decide on the nature of the benchmarking data sets, such that they are both realistic and interesting, (iii) to decide on the specific challenges and (iv) applied evaluation metrics, such that they both are informative for real world applications and accepted by the developer community, as well as (v) to ensure reproducibility of the tool submissions, data sets and the performance evaluation.

3.9 Upcoming challenges in phylogenomics

Siavash Mirarab (University of California at San Diego, US)

License  Creative Commons BY 3.0 Unported license
© Siavash Mirarab

A major challenge in reconstructing evolutionary histories (i.e., phylogenies) is accounting for the potential discordance between histories of individual genes (i.e., gene trees) and the species as a whole (i.e., the species tree). Reconstructing phylogenies from genome-scale data has the promise to address this long-standing challenge in phylogenetics. However, several new challenges are presented when genome-wide data are used for phylogeny inference. At the highest level, the definition of a gene and a species becomes important and non-trivial. Scalable methods for species delimitation and for selecting recombination-free regions of the

genome are needed; moreover, we need to better understand impacts of recombination on phylogeny estimation, both at the gene and the species level. Simultaneous modeling of multiple causes of discordance between gene trees and the species tree is also challenging, both from theoretical and practical perspectives. When models that incorporate multiple causes of discordance are designed, inference under them often becomes an intractable computational problem. This has limited the best of existing methods that handle multiple causes of discordance to no more than tens of species. Finally, testing the accuracy of genome-scale phylogenies and interpreting the results generated by various methods requires care; improved methods for assessing support will be needed.

3.10 Recent advances and future challenges in BWT

Laurent Mouchard (University of Rouen, FR)

License  Creative Commons BY 3.0 Unported license
© Laurent Mouchard


Given a text T , the Burrows-Wheeler Transform of T is the last column of the conceptual matrix where rows are alphabetically ordered cyclic shifts of T . $BWT("BANANA\$")=ANNB\AA . This reversible transform, that does not compress text has a tendency of aggregating similar individual letters. It has been used as a preprocessing tool for compressors such as bzip2 for example. There exists a function, named LF (Last-First) that can be used for recovering the original text T when one has only access to $BWT(T)$. This transform and the corresponding data structure has been used in the context of Next-Generation Sequencing for preprocessing the reference sequences in order to speed up the detection of starting positions of myriads of short fragments (reads) in the reference sequences. Some technical aspects, such as time and space complexity are addressed. Several recent advances are presented:

- Dynamic and relative BWT
- Role of BWT in the context of FM-indices
- BWT of a set of highly similar sequences
- BWT construction using external memory
- Merging BWTs

A brief overview of future challenges is presented that paves the way for interactions/discussions during the Seminar.

3.11 Examples where theory fails in practice and practice needs some theory

Gene Myers (MPI – Dresden, DE)

License  Creative Commons BY 3.0 Unported license
© Gene Myers

We present a number of examples in the area of noisy, long read DNA reconstruction (assembly) where theory fails in practice:

- BWT's are theoretically superior, but k-mer sort and merge provides faster read mapping and overlap.
- Current multi-alignment heuristics are too slow and unable to separate polyploid genomes.

- We suggest that graphs are not good representations for pan genomics as they give unintuitive representation of next reversals, transpositions, and inversions.

And where practice needs some theory:

- A clear elucidation of the differences between deBruijn and string graphs is needed, along with an understanding of the limitation of each.
- CIGAR notation for alignments is space inefficient for noisy reads, and current formats are not designed for simplicity of adoption and machine reading.
- We suggest that assembly benchmarking would be significantly more informative if simulated data and theoretical sound metrics were used.
- HPC middle-ware is cumbersome and not tailored to bioinformatics.
- Good visualization and editing tools for assemblies still do not exist.

3.12 Pangenome Variant Calling

Veli Mäkinen (University of Helsinki, FI)

License  Creative Commons BY 3.0 Unported license
© Veli Mäkinen

Detection of genomic variants is commonly conducted by aligning a set of reads sequenced from an individual to the reference genome of the species and analyzing the resulting read pileup. Typically, this process finds a subset of variants already reported in databases and some novel mutations characteristic to the sequenced individual. Most of the effort in the literature has been put to the alignment problem on a single reference sequence, although our gathered knowledge on species such as human is pan-genomic: We know most of the common variations in addition to the reference sequence. There have been some efforts to exploit pan-genome indexing, where the most widely adopted approach is to build an index structure on a set of reference sequences containing observed variation combinations.

The enhancement in alignment accuracy when using pan-genome indexing has been demonstrated in experiments, but surprisingly the above multiple references pan-genome indexing approach has never been tested on its final goal, that is, in enhancing variation detection. This is the focus of this article: We study a generic approach to add variation detection support on top of the multiple references pan-genomic indexing approach. Namely, we study the read pileup on a multiple alignment of reference genomes, and propose a heaviest path algorithm to extract a new recombined reference sequence. This recombined reference sequence can then be indexed using any standard read alignment and variation detection workflow. We demonstrate that the approach actually enhances variation detection on realistic data sets.

This is joint work with Daniel Valenzuela, Niko Välimäki, and Esa Pitkänen.

3.13 Challenges of ancient genomics and pan-genomics

Kay Nieselt (Universität Tübingen, DE)


License  Creative Commons BY 3.0 Unported license
© Kay Nieselt

The advent of next-generation sequencing and recently developed enrichment techniques utilizing tailored baits to capture ancient DNA fragments have made it possible to reconstruct

and compare whole genomes of extinct organisms. Computational paleogenomics deals with the reconstruction and analysis of ancient genomes. Ancient DNA has a number of characteristics, such as short fragment lengths (mean length less than 150bp), and damaged bases, which need to be considered when reconstructing the genome, calling SNPs, comparing genomes or reconstructing phylogenies. In each of these four areas I propose several, partly related challenges. The first challenge addresses the question how to optimally reconstruct the genome from short read data. Typically mapping against a modern reference genome is performed, while de novo assembly is rarely possible. Could hybrid solutions be devised? SNP calling from assembled genomes poses a second problem, since often these assembled genomes suffer from low coverage. The third and fourth challenge address the question how to compare ancient and modern genomes. Since one needs a common coordinate system, the question is how to compute whole-genome alignments (WGA) from ancient as well as modern genomes. Or should one rather refrain from WGAs at all? Finally, in the context of phylogeny reconstruction a number of questions remain largely unsolved. One challenge in this area is to compute a lower bound of genome coverage for which a phylogenetic tree can still be reliably built. And finally relating also to the third challenge is the more general question whether phylogenetic trees consisting of modern as well as ancient genomes should be built from WGAs or with alignment-free methods?

3.14 Data structures to employ embeddings of strings under edit distances to vectors under Hamming distance

S. Cenk Sahinalp (Simon Fraser University – Burnaby, CA)

License  Creative Commons BY 3.0 Unported license
© S. Cenk Sahinalp

When comparing or aligning sequences, mismatches are much easier to handle than indels. Recent results in parsing (genomic) strings through random walks based on shared random bits result in a conceptually simple way to embed strings under edit distance to Hamming vectors, approximately preserving their pairwise distances. Such an embedding simplifies the problem of (pairwise or multiple) sequence alignment problem, even though the distortion (in the distance) they imply are higher than what could be tolerated in real world applications.

3.15 Ensembles of HMMs

Tandy Warnow (University of Illinois – Urbana-Champaign, US)

License  Creative Commons BY 3.0 Unported license
© Tandy Warnow

Profile HMMs are a major tool in bioinformatics analyses and are used for multiple purposes, including the representation of multiple sequence alignments, the detection of homology, protein classification, metagenomic taxon identification, protein structure and function prediction, etc. Yet a single profile HMM is not always suitable for representing a large collection of diverse sequences. In this talk, I will present some approaches to representing a collection of aligned sequences using an ensemble of profile HMMs instead of a single profile HMM. These approaches are able to improve phylogenetic placement, large-scale multiple sequence alignment, protein family classification, and metagenomic taxon identification. Not

only do these methods improve on accuracy (precision and recall) compared to methods based on single HMMs, they also provide improved accuracy compared to leading alternative methods. The relevant methods are SEPP (cf. [1]), TIPP (cf. [2]), UPP (cf. [3]), and HIPPI (cf. [4]). The talk is available at <http://tandy.cs.illinois.edu/warnow-dagstuhl.pdf>. The software base is available at <https://github.com/smirarab/sepp> (Siavash Mirarab github page).

References

- 1 Siavash Mirarab, Nam-phuong Nguyen and Tandy Warnow . *SATé-Enabled Phylogenetic Placement*. Proceedings PSB 2012, pp. 247–258, World Scientific, 2012
- 2 Nam-phuong Nguyen, Siavash Mirarab, Bo Liu, Mihai Pop and Tandy Warnow. *TIPP: Taxonomic Identification and Phylogenetic Profiling*. Bioinformatics, Oxford Journals, 2014
- 3 Nam-phuong Nguyen, Siavash Mirarab, Keerthana Kumar and Tandy Warnow . *Ultra-large alignments using Phylogeny-aware Profiles*. Genome Biology, 16:124, BioMed Central, 2015
- 4 Nam-phuong Nguyen, Michael Nute, Siavash Mirarab and Tandy Warnow. *HIPPI: Highly accurate protein family classification with ensembles of HMMs*. To appear, BMC Genomics

3.16 Three problems in metagenomics

Shibu Yooseph (University of Central Florida – Orlando, US)

License © Creative Commons BY 3.0 Unported license
© Shibu Yooseph

Metagenomics is a cultivation independent paradigm that has enabled detailed studies of microbial communities. Sequence data generated from a metagenome sample can be used to make inferences about the taxonomy, genome composition, and metabolic potential of the constituent microbes in the sampled community. However, the nature and volume of data generated by currently used sequencing technologies also pose computational challenges that require the development of efficient algorithms to effectively analyze these data. Here we discuss three computational problems in metagenomics to highlight these challenges and opportunities. First, to improve annotation of databases containing partial protein sequences, we describe approaches that have higher sensitivity than commonly used homology detection methods like BLAST. The higher sensitivity is obtained by combining database sequence searches together with the assembly of relevant overlapping database sequences to improve homology detection. Second, we describe the computational problem of identifying the host bacterial or archaeal sequences of a given set of viral metagenome sequences, and bottlenecks with current approaches. Third, we consider the problem of developing a unified framework for the estimation of both species abundance curves and metagenome coverage from a set of metagenomic reads.

4 Working groups

4.1 Single-cell cancer genomics: variant calling & phylogeny

Niko Beerenwinkel (ETH Zürich – Basel, CH), Mohammed El-Kebir (Brown University – Providence, US), Gunnar W. Klau (CWI – Amsterdam, NL), Tobias Marschall (Universität des Saarlandes, DE), and S. Cenk Sahinalp (Simon Fraser University – Burnaby, CA)

License © Creative Commons BY 3.0 Unported license
 © Niko Beerenwinkel, Mohammed El-Kebir, Gunnar W. Klau, Tobias Marschall, and S. Cenk Sahinalp

4.1.1 Topics

- Variant calling in single-cell tumor sequencing
 - Single-nucleotide variants (SNV)
 - Copy-number variants (CNV)
 - Structural variants (SV)
 - Phylogeny inference given single-cell tumor sequencing data

4.1.2 Background

- Intra-tumor heterogeneity:
 - Tumor is heterogeneous composed of different cell populations with different somatic mutations.
 - With bulk sequencing the observations are a composite signal from different cell populations => requiring deconvolution
 - This is not the case with single-cell sequencing (SCS) where the observations are from a single cell
- There are specific errors with SCS due to the whole-genome amplification (WGA) step
 - High false negative rate in SNV calling due to allele drop-out in the WGA step
 - * Used to be ~40%; now improved to ~10%
 - Elevated false positive rate in SNV calling due to WGA step
 - Non-uniform read coverage
 - More GC-bias
- Single-cell sequencing is becoming more affordable.
 - Right now about 50 cells are sequenced
 - Most SCS studies are done using whole-exome sequencing (non-uniform read coverage is an even bigger issue in this case)

4.1.2.1 Questions

- Has reproducibility of single-cell sequencing been studied?
 - Nick Navin studied this in healthy cells

4.1.3 SNV calling

4.1.3.1 Issues

- Noisy data with high FP and FN rate (see Background).

4.1.3.2 Approaches

- Use SNV callers that were designed for bulk-sequencing (GATK, MuTect, ...)
- New SNV caller specific for SCS data: Monovar
 - Accounts for allele drop-out and elevated FP rate
 - Uses dynamic programming to compute posterior probabilities and to call SNVs for each cell with max posterior probability.
- Phylogeny inference under the infinite sites assumption to clean up noisy observations: SCITE and OncoNEM.

4.1.3.3 Opportunities

- Do SNV calling by considering all sequenced single-cells of a tumor simultaneously.
 - Monovar is considering cells one by one (with respect to the normal), i.e. assuming independence of cells
- Do SNV calling jointly with phylogeny inference
- Do SNV calling by integrating bulk-sequencing samples.

4.1.4 CNV calling

4.1.4.1 Issues

- Non-uniform read coverage
- Most SCS data is whole-exome only

4.1.4.2 Approaches

- Ginkgo

4.1.4.3 Opportunities

- Joint inference of all cells simultaneously
- In the context of a phylogeny?

4.1.5 Phylogeny inference

4.1.5.1 Motivation

Why do we care about the tree?

- To quantify heterogeneity
- To study the evolutionary process in cancer: is it a burst or is it gradual?
 - Neutral evolution model: (Big Bang): star phylogeny
 - Clonal expansion model: non-star phylogeny
 - These hypotheses can be tested.
- To study the trees of a cohort of patients where we have phenotype and treatment information.
 - Can we find patterns in the trees related to a phenotype?
- To study metastases and migration of tumor cells
 - Where do tumor cells that circulate in the blood come from?
 - Oliver raises the point that in melanoma the metastasis are different from the primary tumor.

4.1.5.2 Approaches

- SCITE
- OncoNEM

4.1.6 Ideas

- Combining bulk and single-cell sequencing
 - How many single cells do you need to sequence in order to detect all relevant clones?
 - * Should we sequence all billion cells of a tumor?
 - * This is a sampling question and it depends on the tumor being well-mixed, and whether there are selective advantages.
- Can we get time-series data?
 - Liver cancer is a candidate:
 - * It's not surgically removed and thus time-series samples can be obtained by a needle while the patient is under treatment
 - * Niko says this is painful and thus hard to get such samples, but Oliver may have access to such samples.
 - Leukemia
- What is a good generative model for the somatic mutational process in cancer?
 - This will allow us to validate variant calling and phylogeny inference methods.
 - Niko suggests that HMMs are enough
 - Tandy prefers phylo-HMMs [refs] or tree-based HMMs [refs]
- Philosophical discussion about Bayesian approaches
 - Niko: The following is a misconception: Bayesian computations are expensive, and likelihood computations are cheap.
 - * In some cases sampling from the posterior is hard to achieve
 - * It takes a long time for the MCMC chain to mix
 - Max Likelihood approaches: You can do anything to optimize the objective function.
 - Bayesian inference: Any sampling schemes that construct a proper Markov chain are fine. Some converge faster than others. Anything goes.
 - Bayesian: How to summarize your posterior?
 - How to communicate the uncertainty?

4.2 Cancer genomics

Mohammed El-Kebir (Brown University – Providence, US), Niko Beerenwinkel (ETH Zürich – Basel, CH), Christina Boucher (Colorado State University – Fort Collins, US), Anne-Katrin Emde, Birte Kehr, Gunnar W. Klau (CWI – Amsterdam, NL), Pietro Lio' (University of Cambridge, GB), Siavash Mirarab, Luay Nakhleh, Esko Ukkonen (University of Helsinki, FI), and Tandy Warnow (University of Illinois – Urbana-Champaign, US)

License © Creative Commons BY 3.0 Unported license

© Mohammed El-Kebir, Niko Beerenwinkel, Christina Boucher, Anne-Katrin Emde, Birte Kehr, Gunnar W. Klau, Pietro Lio', Siavash Mirarab, Luay Nakhleh, Esko Ukkonen, and Tandy Warnow

Cancer is a disease caused by somatic mutations that accrue in a population of cells during the lifetime of an individual [6]. This process can be described by a phylogenetic tree and results in different subpopulations of cells, or *clones*, each with different complements of somatic mutations. A clone is composed of all cells that share the same most recent common ancestor,

or equivalently all the leaves that occur in a subtree of the phylogeny. This definition of a clone is elusive: at one extreme all tumor cells form a clone, whereas at the other extreme each tumor cell is a clone. The desired resolution is not clear and depends on the specific application.

Here, we discuss recent trends in computational cancer genomics and identify topics of interest with open computational challenges.

4.2.1 Bulk vs. Single-cell Sequencing

Most cancer sequencing studies are performed using bulk-sequencing technology, where the observations are composite signals from a mixture of cells with different somatic mutations. In contrast, with single-cell sequencing (SCS) the observations are from individual tumor cells. However, there are specific errors that occur during the whole-genome amplification (WGA) step, including segmental drop out where not all copies of a genomic segment are amplified. The used sequencing approach has thus implications in variant calling and phylogeny inference, and requires tailored methods and error models as we will discuss in the following.

4.2.2 Variant Calling

Somatic variants differ in size and include single-nucleotide variants (SNVs) that affect individual genomic positions, copy-number variants (CNVs) that affect larger genomic regions and more complex structural variants (SVs) that do not necessarily change the copy number such as inversions. Calling somatic SNVs and CNVs in tumor bulk-sequencing samples with respect to a matched normal samples requires dealing with mixed samples, where variants do not necessarily occur in all cells. This topic has been studied extensively in the literature but it is not solved and remains a hard problem. Here, we focus on variant calling in SCS data where we have to account for SCS-specific errors.

In the context of SNV calling, allele drop-out leads to elevated false positive and false negative rates. For instance, not observing any reads with an SNV does not mean that the SNV is not present in a tumor cell as the segment containing the SNV could simply have failed to amplify in the WGA step. Recently, the method Monovar has been proposed for calling SNVs that accounts for errors specific to SCS [10].

Typically, read depth is used to infer copy-number values for genomic segments. However, due to allele drop-out, read depth is non-uniform in SCS data even for healthy cells that are heterozygous diploid. This effect is even more pronounced in whole-exome sequencing data where only 3% of the genome is sequenced. There are thus several opportunities in calling SNVs and CNVs for SCS data. For instance, considering all tumor cells simultaneously could improve consistency in the calls. Moreover, joint phylogeny inference and calling may further improve the accuracy.

4.2.3 Structural Variation Calling

Calling of structural variants (events > 50 bp) poses some additional challenges. Short read technologies are inherently limited in their capability to detect SVs, especially when events are complex and involve repetitive sequences. Moreover, wet-lab validation of such events can be difficult and even for germline SVs no comprehensive ground truth data sets are available. Therefore, biological formation mechanisms are far from being fully understood.

Most algorithms that act on short read data use a combination of read-pair, split-read, and read-depth signals and also several local assembly approaches have been developed. However, different tools can lead to very different SV call sets [9]. And even when tools

agree, combining the different types of signals into robust variant-allele fraction estimates is non-trivial [3].

Long-range technologies, including long reads, synthetic long reads and optical mapping, have the potential of resolving SVs better, especially when combined with short read data and when used to infer SVs simultaneously or iteratively with CNVs [1]. Another potential opportunity might be combining long read data with improved single-cell technology to monitor accumulation of variants over time, which could lead to a deeper understanding of SV formation.

While exploring these opportunities, better visualization tools (such as e.g. GenomeRibbon¹) and more consistent file formats for SV calls will be needed in order to make the calls more accessible and easier to handle.

4.2.4 Phylogeny Inference

Inferring tumor phylogenies allows one to study and test the applicability of different modes of evolution in human cancers such as the clonal expansion model [5] or the Big Bang model [8]. Moreover, studying phylogenetic trees of a cohort of patients where we have phenotypic and treatment information allows one to identify patterns that are related to specific phenotypes or treatment.

There are several challenges in phylogeny inference depending on the used sequencing strategy. SCITE [4] infers phylogenetic trees using SNVs under the infinite sites assumption and uses a likelihood model to account for elevated FP and FN rates in SCS data. Studying whether the infinite sites assumption is a reasonable assumption, especially in the context of copy-number variants, is an interesting open question. With bulk-sequencing data, tree inference methods must account for mixed samples and simultaneously solve a deconvolution and tree inference problem [2]. Since variant allele frequencies of SNVs are confounded by CNVs, it is thus essential to jointly consider SNVs and CNVs – and ideally all types of variants including SVs – when inferring phylogenetic trees given bulk-sequencing data.

4.2.5 Integrative Analysis

Integrated analysis of different molecular profiles obtained from the same tumor allows one to comprehensively study a tumor. For instance combined expression (RNA-seq) and mutation (DNA-seq) data may improve variant calling and could allow one to study the effect of somatic variants on expression, including alternative splicing and gene fusions. Moreover, combining different sequencing strategies could mitigate the challenges associated to the individual strategies and lead to an over-all better understanding of the somatic variants present in a tumor and their evolutionary history.

All of the above mentioned challenges require the development of novel methods. However, subsequent validation of such methods is difficult as no ground truth is available. Hence, it is also essential to formulate good generative models that comprehensively capture the somatic mutational process in cancer. Such models are currently missing, and we propose to consider and adapt existing models used in species evolution [7].

References

- 1 Xiang Chen, Pankaj Gupta, Jianmin Wang, Joy Nakitandwe, Kathryn Roberts, James D. Dalton, Matthew Parker, Samir Patel, Linda Holmfeldt, Debbie Payne, et al.

¹ <http://www.genomeribbon.com>

- CONCERTING: integrating copy-number analysis with structural-variation detection. *Nature Methods*, 12:527–530, 2015.
- 2 Mohammed El-Kebir, Gryte Satas, Layla Oesper and Benjamin J. Raphael *Inferring the Mutational History of a Tumor Using Multi-state Perfect Phylogeny Mixtures*. *Cell Systems*, 3(1):43–53, July 2016.
 - 3 Xian Fan, Wanding Zhou, Zechen Chong, Luay Nakhleh and Ken Chen . Towards accurate characterization of clonal heterogeneity based on structural variation. *BMC Bioinformatics*, 15(1):1–12, 2014.
 - 4 Katharina Jahn, Jack Kuipers and Niko Beerenwinkel. *Tree inference for single-cell data*. *Genome biology*, 17(1):86, May 2016.
 - 5 Serena Nik-Zainal et al. *The life history of 21 breast cancers*. *Cell*, 149(5):994–1007, May 2012.
 - 6 P C Nowell . *The clonal evolution of tumor cell populations*. *Science*, 194(4260):23–8, Oct 1976.
 - 7 Adam Siepel and David Haussler . *Combining phylogenetic and hidden Markov models in biosequence analysis*. *Journal of Computational Biology*, 11(2-3):413–428, 2004.
 - 8 Andrea Sottoriva, Haeyoun Kang, Zhicheng Ma, Trevor A. Graham, Matthew P. Salomon, Junsong Zhao, Paul Marjoram, Kimberly Siegmund, Michael F Press, Darryl Shibata and Christina Curtis. *A Big Bang model of human colorectal tumor growth*. *Nature Genetics*, 47(3):209–216, March 2015.
 - 9 Peter H. Sudmant, Tobias Rausch, Eugene J. Gardner, Robert E. Handsaker, Alexej Abyzov, John Huddleston, Yan Zhang, Kai Ye, Goo Jun, The 1000 genomes project consortium, et al. An integrated map of structural variation in 2,504 human genomes. *Nature*, 526:75–81, 2015.
 - 10 Hamim Zafar, Yong Wang, Luay Nakhleh, Nicholas Navin and Ken Chen . *Monovar: single-nucleotide variant detection in single cells*. *Nature methods*, 13(6):505–507, May 2016.

4.3 Software libraries for indexing

Simon Gog (KIT – Karlsruher Institut für Technologie, DE), Pascal Costanza (Intel Corporation, BE), Anthony J. Cox (Illumina – United Kingdom, GB), Fabio Cunial (MPI – Dresden, DE), Hannes Hauswedell (FU Berlin, DE), André Kahles (ETH Zürich, CH), Ben Langmead (Johns Hopkins University – Baltimore, US), Laurent Mouchard (University of Rouen, FR), Gene Myers (MPI – Dresden, DE), Enno Ohlebusch (Universität Ulm, DE), Simon J. Puglisi (University of Helsinki, FI), Gunnar Rättsch (ETH Zürich, CH), Knut Reinert (FU Berlin, DE), Bernhard Renard (Robert Koch Institut – Berlin, DE), Enrico Siragusa (IBM TJ Watson Research Center – Yorktown Heights, US), German Tischler (MPI – Dresden, DE), and David Weese (SAP Innovation Center – Potsdam, DE)

License © Creative Commons BY 3.0 Unported license

© Simon Gog, Pascal Costanza, Anthony J. Cox, Fabio Cunial, Hannes Hauswedell, André Kahles, Ben Langmead, Laurent Mouchard, Gene Myers, Enno Ohlebusch, Simon J. Puglisi, Gunnar Rättsch, Knut Reinert, Bernhard Renard, Enrico Siragusa, German Tischler, and David Weese

4.3.1 Topics

- Recent work on bidirectional BWT
- Future plans to build Seqan on top of sdsl-lite
- Understanding why people use/don't use Seqan or other libraries?

4.3.2 Areas of interest – why are we here?

- Andre: interested in what is out there
- Enno: pan genome, compressed de Bruijn graphs, understanding what else people want
- Laurent: dynamic data structures, pangenomes
- Ben: latest on Seqan, sdsl, pangenomes
- Tony: indexing variant sets
- Enrico: SDSL, pangenomes, good implementations of assembly graphs
- David: k-mer index
- German: indexing large data structures, semi-external methods
- Pascal: efficiency of implementation
- Knut: practicalities of integrating libs

4.3.3 Knut – recent work on 2-directional BWT

- Two FM indexes – fwd and reverse, back in 1 = fwd in other
- Need cardinality of intervals
- Originally by Lam, improved by Ohlebusch, Gog + others Prefix sums would speed up but need to store them Present work shows – can do this with only extra bit per BWT character
- Simon believes can get rid of bit too
- bitwise ops Implemented in Seqan 7.44s → 4.79s over wavelet tree on DNA Space e.g. 88Mb → 131Mb
- See preprint for more detail: <https://arxiv.org/abs/1608.02413>
- German – implement fwd and reverse complement in 1 index, searches both dirs in 1 search, as done by Heng Li
- Ben – may still need rev index to do approx matching/branching? German believes not the case
- Discussion around how Knut's data structure would go into sdsl-lite
- Simon – self-contained data structures – support structures – augment self-contained ds (and has pointer to it), eg add rank support
- further discussion

4.3.4 reuse vs rewrite

- Solution based on minimal perfect hash functions? (from Veli Makinen) Simon – MPHFs popular in information retrieval but not in bioinformatics
- Discussion on space usage

4.3.5 Latest and greatest

- German (libmaus2 on github)
 - Huffman and RLE
 - own alg for indexing DNA, scales up to NCBI ref db, 1.5Tb inc fwd and r/c on github
 - GPL, mandated by Sanger (not ideal for integ with Seqan)
 - alg published but not exptl results
- Ben
 - Bowtie uses own implementation of FM index
 - Index building – Burkhardt + Karkkainen, parallelized
 - Bowtie2 uses Lam's bidirectional index

- Simon on SDSL
 - support for small and big alphabets (1 million) – latter needed for IR apps
 - inspiration from Pizza/Chili – generic imps of common components
 - bit vectors, rank/select
 - 8 flavours of wavelet tree! (choice depends on alphabet size among other factors)
 - 2 page cheat sheet describes everything
 - parameterize wavelet tree by bit vector – many combinations
 - configured at compile time in the manner of C++ STL
 - not only performance advantages, but also Pizza/Chili was hard to configure at index construction time via the API
 - Polymorphic construct() function builds anything
- Ben – avoid file copy by memory mapping?
- Simon – code to do this on branch right now
- relative not absolute pointers important
- David – is there abstract interface to string so that eg string non-contiguous in memory could be used?
- Large page sizes Configure OS for large page sizes as recommended by SNAP developers
- Kurt on SeqAN
 - Seqan BAM → SAM is 2x faster than htlib
 - Compile-time parameterization by alphabet type and index type (eg q-gram index, FM index)
 - Generic iterator interface for tree traversal
 - Compile-time generic programming module for dynamic programming (192 flavours!)
 - Multiple genomes stored using journaled string tree
 - * Q: can you index this? A: Yes/no!
 - * 15% overhead (of JST) for 1 string but 50-fold speedup for 130
 - * easy to add or delete a sequence
 - Working with Intel to add vectorization to core lib
 - Multithreading
- for SeqAn 3.0
 - Separate apps from core in build system
 - C++14, C++17 features where poss
 - multithread/SIMD of core components
 - external libraries eg sds, maybe graph libs
- Q: CRAM support? A: a lot of overhead to fully implement the CRAM spec Q: use htlib for CRAM? A: probably a lot of overhead in converting internal structs
- pragma simd to force vectorization, was in Intel compiler only but is now in OpenMP
- Recommendations on Cilk vs openMP vs TBB [can someone else summarize please]

Optimizing vectorized code:

 - vector reports compiler switch to see what is vectorized...
 - and these reports can be embedded in source code ...
 - Vtune is GUI for this
 - or use gdb or Intel's equiv to look at assembly language directly

4.4 Assembly

Gene Myers (MPI – Dresden, DE), Jason Chin (Pacific Biosciences – Menlo Park, US), Richard Durbin (Wellcome Trust Sanger Institute – Cambridge, GB), Mohammed El-Kebir (Brown University – Providence, US), Anne-Katrin Emde (New York Genome Center, US), Birte Kehr (deCode Genetics – Reykjavik, IS), Oliver Kohlbacher (Universität Tübingen, DE), Veli Mäkinen (University of Helsinki, FI), Alice Carolyn McHardy (Helmholtz Zentrum – Braunschweig, DE), Laurent Mouchard (University of Rouen, FR), Kay Nieselt (Universität Tübingen, DE), Adam M. Phillippy (National Institutes of Health – Rockville, US), Tobias Rausch (EMBL – Heidelberg, DE), Peter F. Stadler (Universität Leipzig, DE), Granger Sutton (The J. Craig Venter Institute – Rockville, US), German Tischler (MPI – Dresden, DE), and David Weese (SAP Innovation Center – Potsdam, DE)

License © Creative Commons BY 3.0 Unported license

© Gene Myers, Jason Chin, Richard Durbin, Mohammed El-Kebir, Anne-Katrin Emde, Birte Kehr, Oliver Kohlbacher, Veli Mäkinen, Alice Carolyn McHardy, Laurent Mouchard, Kay Nieselt, Adam M. Phillippy, Tobias Rausch, Peter F. Stadler, Granger Sutton, German Tischler, and David Weese

4.4.1 Topics

Proposed discussion topics

- Assembly data format
 - Beyond linear representation
 - Capture ambiguity and quality
- Emerging technologies
 - Optimal/economic integration
 - Genome finishing
- Pre-assembly QC
 - Estimating ploidy, genome size, repeat content from raw reads
 - Error correction
- Population assembly, cancer assembly, metagenome assembly (other group)
- Local assembly for variant detection
- Assembly and graph visualization
- High-performance computing
- Provocation: Why isn't assembly solved? What's missing to solve it?

4.4.2 Assembly data format

A unified data format is needed that captures the full information (and ambiguity) of an assembly

Chin, Durbin, and Myers proposed an extension of the GFA format

- Vertices are segments, edges are overlaps
- Describes consensus and multi-alignments
- Consistent with SAM notations
- Segments can be with or without pieces (specify coordinate + alignment info (cigar or trace))
- Edge types: dovetail, branch, contain
- Expressive enough to describe the full assembly (graph + segments + pieces + alignments)
- History tracking through SAM header

Major proposed changes to the current GFA spec

- 'Pieces' as subcomponents of segments
- 'Branches' as any local alignment between segments
- Optional alignment formats (cigar or trace)
- Object-size prolog (debated)

Questions/Remarks

- Best way to encode haplotypes? With local alignments?
- Enough to have just one link type?
- What about scaffolds with ambiguous gap sizes?
- Provide validator/convertor for new format
- Develop binary version of format
- JSON or SAM style?
- Is there a way to represent collections?
 - All segments of a given chromosome (e.g. from Hi-C clustering)
 - All segments of a given organism (e.g. from metagenomic binning)
- Converters to common/alternate graph formats needed
- Are 'general' edges too flexible?
 - Can now represent all local alignments between segments
 - What does this graph structure represent? How it is visualized?
- NCBI is interested in an assembly submission format. What are their needs/requirements?

Proposed spec GFA 2.0 is here: <https://github.com/thegenemyers/GFA-spec>

4.4.3 Emerging technologies

Technologies for building great assemblies: what's new?

4.4.3.1 Technology list

- Long reads (PacBio, Nanopore)
- Short reads (Illumina)
- Synthetic long reads (Illumina TSLR)
- Linked reads (10x Genomics)
- In vitro Hi-C (Dovetail)
- In vivo Hi-C (PhaseGenomics)
- Optical Maps (BioNano)

4.4.3.2 Considerations

- Different technologies offer different resolution and accuracy
- Economics of best reconstruction (What kind of assembly do you need?)
- Contig vs. scaffold size (PacBio vs. 10x)
- New scaffolding opportunities with chromatin interaction frequency (Hi-C)
- New phasing options (10x, PacBio, Hi-C)
- Complementarity. Where do technologies break? (e.g. PacBio vs. BioNano)
- Iterative improvement and validation using multiple techs
 - E.g. PacBio → BioNano → Hi-C gives chromosome-scale assemblies

4.4.4 Pre-assembly QC

Is it my genome, my data, or my assembler that is causing the problem?

4.4.4.1 Suggestions

- Illumina
 - Compute k-mer frequency to estimate haploid and diploid coverage, repetitiveness, and genome size
 - Might be difficult for Hi-C due to non-uniform coverage
- PacBio
 - Count overlaps, rather than k-mers, to estimate coverage, repeats, and genome size
 - Reads can be too noisy for k-mer based approach

4.5 Big data

Gene Myers (MPI – Dresden, DE), Ewan Birney (European Bioinformatics Institute – Cambridge, GB), Pascal Costanza (Intel Corporation, BE), Anthony J. Cox (Illumina – United Kingdom, GB), Fabio Cunial (MPI – Dresden, DE), Richard Durbin (Wellcome Trust Sanger Institute – Cambridge, GB), Simon Gog (KIT – Karlsruher Institut für Technologie, DE), Hannes Hauswedell (FU Berlin, DE), Birte Kehr (deCode Genetics – Reykjavik, IS), Ben Langmead (Johns Hopkins University – Baltimore, US), Laurent Mouchard (University of Rouen, FR), Enno Ohlebusch (Universität Ulm, DE), Adam M. Phillippy (National Institutes of Health – Rockville, US), Mihai Pop (University of Maryland – College Park, US), Simon J. Puglisi (University of Helsinki, FI), Tobias Rausch (EMBL – Heidelberg, DE), Karin Remington (Computationality, US), S. Cenk Sahinalp (Simon Fraser University – Burnaby, CA), Peter F. Stadler (Universität Leipzig, DE), and German Tischler (MPI – Dresden, DE)

License © Creative Commons BY 3.0 Unported license

© Gene Myers, Ewan Birney, Pascal Costanza, Anthony J. Cox, Fabio Cunial, Richard Durbin, Simon Gog, Hannes Hauswedell, Birte Kehr, Ben Langmead, Laurent Mouchard, Enno Ohlebusch, Adam M. Phillippy, Mihai Pop, Simon J. Puglisi, Tobias Rausch, Karin Remington, S. Cenk Sahinalp, Peter F. Stadler, and German Tischler

Public archives of DNA sequencing data are filled with valuable datasets contributed by projects large and small across the world. They are also growing extremely rapidly; the Sequence Read Archive, for example, has a doubling time of about 18 months. In the days before next-generation sequencing dominated the field, public databases were easy for everyday scientists to query. Today, these databases contain petabytes of data. While simply storing this data has recently become practical and sustainable – thanks in part to improved compression – the task of querying these databases, or even a large fraction thereof, is now very challenging. It's not possible for a typical biological researcher to rapidly query the large archives like the Sequence Read Archive or the European Nucleotide Archive.

We feel that an important focus of Computational Genomics research should be on tackling the problem of indexing very large amounts of sequencing data. Advances in this field would have two major benefits: it would make it easier for typical scientists to query these archives, and it would create an important incentive for producers of “private” sequencing data (e.g. clinical samples) to eventually release them to the public in some form. We also

note that the Global Alliance for Genomics and Health (GA4GH) have proposed mechanisms for allowing limited querying of private sequencing data spread across many loci.

To enable fast queries over archives, the pivotal need is for data structures capable of answering queries that take query sequences and return information about whether and where that queries occur in the raw data. This kind of query is in the spirit of local alignment; while other queries could certainly be useful, we focus on this kind here because it can serve as a building block for many others. We suggest that such a data structure should exist separately from the raw sequencing data; in other words, the raw data would still be stored and made available in an un-indexed form, which, while it does not allow fast queries, does allow a wide variety of methods to be applied. The core data structures we suggest for further investigation are

1. those based on the Burrows-Wheeler Transform (BWT) or FM Index,
2. those based on the de Bruijn graphs,
3. those based on multi-vantage-point trees, and
4. those based on sketching schemes or other schemes that reduce the key space by replacing sequences with representatives that are “nearby” in, say, edit distance space.

These data structures are primarily responsible for finding whether and where sequences occur, but they must be augmented to make it possible to determine which particular archived samples the sequences occur in. This is related to the “document listing problem,” and also related to the “colored” de Bruijn graph.

We briefly attempt to estimate the size required by a colored de Bruijn graph data structure built over a very large archive of sequencing data. We assume that the k -mer length is long enough that the number of distinct keys is governed by the amount of data rather than by the limited number of k -mers. We assume the number of distinct k -mers occurring in the archived data is 10^{12} , and that the number of bits required to associate metadata (i.e. the “color” bits) with each k -mer is about 10^8 per key. This leads to an estimate of about 10^{20} total bits, with additional space needed to store the keys themselves. However, there are many opportunities for compression, since

1. overall, the total collection of bit vectors is sparse; mostly 0s, given that most k -mers are absent from most datasets,
2. if the positions of the bit vectors correspond to samples then there is a dependence structure among the columns; since some samples are biologically similar, we expect them to be similar in k -mer composition, and
3. the bit vectors themselves are dependent since two k -mers that overlap by $k-1$ positions are likely to occur in similar patterns of database samples.

The assumption that only 10^{12} keys are needed needs further discussion. The number is almost certainly much larger in practice when real sequencing data is used. This is because of sequencing errors, which give rise to a very large number of k -mers that are made unique (or nearly so) by the random sequencing errors they overlap. We suggest that some degree of “smoothing” or error correction is needed to reduce the size of the key space prior to building the data structure.

4.6 Structural Variant Detection

Gene Myers (MPI – Dresden, DE), Jason Chin (Pacific Biosciences – Menlo Park, US), Mohammed El-Kebir (Brown University – Providence, US), Anne-Katrin Emde (New York Genome Center, US), Birte Kehr (deCode Genetics – Reykjavik, IS), Veli Mäkinen (University of Helsinki, FI), Tobias Marschall (Universität des Saarlandes, DE), Adam M. Phillippy (National Institutes of Health – Rockville, US), Mihai Pop (University of Maryland – College Park, US), Karin Remington (Computationality, US), S. Cenk Sahinalp (Simon Fraser University – Burnaby, CA), and Granger Sutton (The J. Craig Venter Institute – Rockville, US)

License © Creative Commons BY 3.0 Unported license

© Gene Myers, Jason Chin, Mohammed El-Kebir, Anne-Katrin Emde, Birte Kehr, Veli Mäkinen, Tobias Marschall, Adam M. Phillippy, Mihai Pop, Karin Remington, S. Cenk Sahinalp, and Granger Sutton

4.6.1 Structural Variant Calling

4.6.1.1 Basics

- Definition of SV: variant >50bp
- Types of sequencing-based signals/approaches:
 - Split reads (SR)
 - Read pairs (RP)
 - Read depths (RD)
 - Assemblies
- Challenges for SV calling
 - need for improved SV detection methods
 - need for improved annotation/resources for SVs
 - need for improved file formats and visualization tools
 - lack of biological understanding

4.6.1.2 SV detection methods/algorithms

- filtering of FPs, especially de-novo SVs; LD can help in population data
- current methods have limitations:
 - mostly limited to Illumina PE data, need to integrate technologies
 - most established methods focus on accessible genome (unique reads) but SVs are often in repeats
 - need for better quality scores (both for mapped reads as input into SV calling, and for called SVs)
 - problem of false positives, low validation rates esp. for de-novo SVs
- SV validation difficult
 - intersection of tools as proxy for precision, but shared artifacts as well as true unique calls
 - need for benchmarking data, SV gold standard
 - wetlab validation not straight-forward (cloning vector approaches? longer read techs)
 - SVs that agree with protein (mass spec)
 - SVs are often complex: mini events around SV breakpoints, homologies
- better merging/overlapping of SV sets, removing redundancy
 - resolution differs by method (assembly/split-read > read pair > read depth)
 - merging/comparing is a difficult problem (reciprocal overlap not sufficient)

- germline SV filtering:
 - family structure helps: jointly assess parents with child
 - population structure/haplotype information: LD

4.6.1.3 Need for improved annotation/resources for SVs

- need for useful databases of SVs
 - need for a dbSNP for SVs
 - DGV problematic
 - 1000G is not a great resource yet
- need comprehensively characterized genomes, SV goldstandard
 - currently being analyzed: 3 trios with 10x and strand-seq eventually
- need for improved annotation/resources for SVs
 - prediction of functional impact of SVs

4.6.1.4 Need for improved file formats and visualization tools

- SV visualization tools
 - IGV (limitations): Jason Chin working with IGV folks to add signs for large insertions
 - genomeribbon.com
 - Circos (but static, for complex events)
 - SVviz
- formats: VCF, bedpe, separate format for genotyping

4.6.1.5 Lack of biological understanding

- SV type classification not straight-forward
- mechanisms of SV formation not well-understood
- can knowledge about biological mechanism aid methods for detection?

4.6.1.6 Papers of interest

- Paper from 1000G SV group (mechanisms of SV formation) [1]
- Resolving the complexity of the human genome using single-molecule sequencing [2]
- GoNL de-novo SV [3]
- Veli: paper on merging SVs (tandem repeat regions, deletions, only pairwise)
- BreakDown for SV VAF estimation in cancer [4]

4.6.2 Somatic SV calling

4.6.2.1 What is different in cancer, what is different in single-cells?

Differences of the problem:

- Heterogeneity needs to be taken into account
 - Lower support for variants when analyzing mixtures of subclones (purity)
 - Intersecting variants from different subclones
- Non-uniform coverage in single-cell sequencing
- Comparative approaches: tumor vs normal

Methodological difference:

- Often the same methods as for germline SV detection

- Different post-processing
- Instead of population-wide calling, tumor/normal joint calling

4.6.2.2 Purity estimation / allelic frequency / proportionality problem

- Depends on copy number (purity and copy number can be traded for each other)
- Approach: Joint probability distribution from tumor and normal e.g. 0 in normal, >0 in tumor (HitSeq 2016)

4.6.2.3 Integration of copy-number estimation (RD) and adjacencies (SR + RP) in tools

- JABBA (unpublished): both predicted at the same time
- CONSERING: both predicted iteratively

4.6.2.4 How can you validate calls? How can we get ground truth?

- Idea (Mohammed): Construct tumor phylogeny from SNVs and from SVs separately and compare the result.
- Simulation impossible as long as we don't understand what is going on.
- Single-cell data solves clonality problem, long reads resolve complex events

4.6.2.5 Can we disentangle complex events, can we define atomic event and can we resolve an order in which events have occurred?

- Biologically, events are often more complex than atomic operation (not just simple deletions, insertions, inversions, ...). Fuzzy definition of breakpoints doesn't help.
- Ideal definition: Assuming that we have all cancer cells sequenced, an event is a change between two adjacent cells.
- Question: Is there a clear pattern of these events so that we can define a set of atomic operations?

References

- 1 Alexej Abyzov, Shantao Li, Daniel Rhee Kim, Marghoob Mohiyuddin, Adrian M. Stütz, Nicholas F. Parrish, Ximmeng Jasmine Mu, Wyatt Clark, Ken Chen, Matthew Hurles, Jan O. Korbel, Hugo Y. K. Lam, Charles Lee and Mark B. Gerstein. *Analysis of deletion breakpoints from 1,092 humans reveals details of mutation mechanisms*. Nature Communications 6, 7256, 2015
- 2 Mark J. P. Chaisson, John Huddleston, Megan Y. Dennis, Peter H. Sudmant, Maika Malig, Fereydoun Hormozdiari, Francesca Antonacci, Urvashi Surti, Richard Sandstrom, Matthew Boitano, Jane M. Landolin, John A. Stamatoyannopoulos, Michael W. Hunkapiller, Jonas Korlach and Evan E. Eichler. *Resolving the complexity of the human genome using single-molecule sequencing*. Nature, 517, pp. 608–611, 2015
- 3 Wigard P. Kloosterman, Laurent C. Francioli, Fereydoun Hormozdiari, Tobias Marschall, Jayne Y. Hehir-Kwa, Abdel Abdellaoui, Eric-Wubbo Lameijer, Matthijs H. Moed, Vyacheslav Koval, Ivo Renkens, Markus J. van Roosmalen, Pascal Arp, Lennart C. Karssen, Bradley P. Coe, Robert E. Handsaker, Eka D. Suchiman, Edwin Cuppen, Djie Tjwan Thung, Mitch McVey, Michael C. Wendl, Genome of the Netherlands Consortium, André Uitterlinden, Cornelia M. van Duijn, Morris A. Swertz, Cisca Wijmenga, GertJan B. van Ommen, P. Eline Slagboom, Dorret I. Boomsma, Alexander Schönhuth, Evan E. Eichler, Paul I.W. de Bakker, Kai Ye and Victor Guryev. *Characteristics of de novo structural changes in the human genome*. Genome Research, 25, pp. 792–801, 2015.

- 4 Xian Fan, Wanding Zhou, Zechen Chong, Luay Nakhleh and Ken Chen. *Characteristics of de novo structural changes in the human genome*. BMC Bioinformatics, 15:299, BioMed Central, 2014

4.7 Visualization Group

Gene Myers (MPI – Dresden, DE), Jason Chin (Pacific Biosciences – Menlo Park, US), Mohammed El-Kebir (Brown University – Providence, US), Anne-Katrin Emde (New York Genome Center, US), Birte Kehr (deCode Genetics – Reykjavik, IS), Veli Mäkinen (University of Helsinki, FI), Tobias Marschall (Universität des Saarlandes, DE), Adam M. Phillippy (National Institutes of Health – Rockville, US), Karin Remington (Computationality, US), S. Cenk Sahinalp (Simon Fraser University – Burnaby, CA), and Granger Sutton (The J. Craig Venter Institute – Rockville, US)

License © Creative Commons BY 3.0 Unported license
© Gene Myers, Jason Chin, Mohammed El-Kebir, Anne-Katrin Emde, Birte Kehr, Veli Mäkinen, Tobias Marschall, Adam M. Phillippy, Karin Remington, S. Cenk Sahinalp, and Granger Sutton

God created visualization and he saw it was good.

We focused on assembly/pan-genome visualization. The first question is defining the purpose – what do we want to visualize and what are the question we want to answer with them.

One observation is that in pan-genomes there are chunks of conserved regions interspersed by highly variable regions. We don't have a good way of visualizing the highly variable region, or interpreting its content in the context of its neighborhood. Some relevant questions may be: are there genes disrupted by this region?; are there specific variants? etc.

These problems are much easier to conceptualize in the context of pan-genomes rather than metagenomic assembly graphs. In assembly graphs complexity due to repeats and errors cannot be easily distinguished from actual biological signals (translocations, strain variants).

Finding the tangles in the graph may be attempted by using the SPQR tree datastructure that hierarchically decomposes a bi-connected graph into tri-connected components (the tangles/variants). In the pan-genome setting this may be achieved with simpler algorithms.

We discussed the Pan-Tetris paradigm (cf. [1]) that is gene-centric and also models the ordering of genes. The visual representation makes it easy to 'combine' tracks representing orthologous genes that may have been mis-aligned in the multiple alignment or have mis-annotated. An important functionality not present is ability to use this information to edit and update the underlying genome alignment or annotation.

In terms of updates we discussed the importance of consistency checks and version tracking to prevent and enable recovery from errors.

We also discussed the need for hierarchical visualizations (SPQR trees for example can provide such a mechanism for assembly graphs) going from the large structure down to the base level.

References

- 1 André Hennig, Jörg Bernhardt and Kay Nieselt *Pan-Tetris: an interactive visualisation for Pan-genomes*. BMC-Bioinformatics, 16(Suppl 11), BioMed Central, 2015

4.8 Metagenomics

Mihai Pop (University of Maryland – College Park, US), Pascal Costanza (Intel Corporation, BE), Anthony J. Cox (Illumina – United Kingdom, GB), Fabio Cunial (MPI – Dresden, DE), Simon Gog (KIT – Karlsruher Institut für Technologie, DE), Hannes Hauswedell (FU Berlin, DE), Daniel H. Huson (Universität Tübingen, DE), André Kahles (ETH Zürich, CH), Pietro Lio' (University of Cambridge, GB), Alice Carolyn McHardy (Helmholtz Zentrum – Braunschweig, DE), Siavash Mirarab (University of California at San Diego, US), Kay Nieselt (Universität Tübingen, DE), Enno Ohlebusch (Universität Ulm, DE), Simon J. Puglisi (University of Helsinki, FI), Gunnar Rätsch (ETH Zürich, CH), Karin Remington (Computationality, US), Bernhard Renard (Robert Koch Institut – Berlin, DE), Enrico Siragusa (IBM TJ Watson Research Center – Yorktown Heights, US), Tandy Warnow (University of Illinois – Urbana-Champaign, US), and Shibu Yooseph (University of Central Florida – Orlando, US)

License © Creative Commons BY 3.0 Unported license

© Mihai Pop, Pascal Costanza, Anthony J. Cox, Fabio Cunial, Simon Gog, Hannes Hauswedell, Daniel H. Huson, André Kahles, Pietro Lio', Alice Carolyn McHardy, Siavash Mirarab, Kay Nieselt, Enno Ohlebusch, Simon J. Puglisi, Gunnar Rätsch, Karin Remington, Bernhard Renard, Enrico Siragusa, Tandy Warnow, and Shibu Yooseph

4.8.1 Topics Proposed for Discussion

Originally we proposed the following topics for additional discussion.

- Taxonomic analysis
- Analyses of viruses, fungi, Eukaryotes . . .
- Functional analysis
- (Metagenome) Assembly
- Strain reconstruction
- How do you want to benchmark
- Integration of -omics data

In the end, the discussion focused on the many challenges posed by the first set of topics and we did not discuss issues surrounding the integration of multiple types of omics data. Also, issues related to databases were found to be central to multiple of the topics. A summary of the discussions is provided below.

4.8.2 Taxonomic issues

While bacterial taxonomy was historically based on morphology, taxonomic schemes have largely moved to including the use of molecular sequence data to organize bacteria (and archaea). However, given that events like lateral gene transfer are common among bacterial groups, it is often also problematic to represent bacterial evolution and relationships using a tree structure. In addition, higher resolution of bacterial groups are complicated by the current lack of formal definitions (i.e. with mathematical utility) for concepts like “bacterial strains”. Due to these difficulties, the NCBI database has stopped tracking the concept of “strain” altogether. We discussed whether this is a problem – after all the strain sequences would have the nucleotide sequence ID as an identifier making it unnecessary to “pollute” the taxonomy database as well. We decided that in some cases having a definition of strain may be useful. One option is to create a domain-specific strain labeling scheme, e.g., *Staphylococcus aureus* MEC+ for a *S. aureus* strain containing the MEC cassette. For consistency, this annotation should be defined computationally and may only make sense within a specific domain. As such a same strain may acquire different “names” in the context of antibiotic

resistance, as opposed to its annotation in the context of bioremediation, for example. A translation between the many possible naming schemes could also be defined computationally, or better yet, the sequence of the strain could represent the ultimate identifier linking the different databases.

In the discussion of taxonomy we also noted that official naming rules are unnecessarily strict – an organism must be isolated, assayed for a number of biological attributes, given a name that follows valid Latin grammar, and submitted and accepted to the International Journal of Systemic and Evolutionary Microbiology (a fun read on the topic is at <http://biorxiv.org/content/biorxiv/early/2016/01/19/037325.full.pdf>).

While computational representations for a taxonomy (e.g., the names.dmp, nodes.dmp representation of the NCBI taxonomy) are preferable for computational analyses, we discussed the need for real names for taxonomy labels as these names are meaningful for biologists. A number of taxonomies (RDP, GreenGenes, SILVA) impose arbitrary restriction on the number of levels of a taxonomy simply due to computational convenience – it is easy to parse names into levels by splitting strings (the textual representation of a path in a tree) only if the number of levels is consistent. We argued for a more expressive computational representation, coupled with an optional textual representation of the paths which would only be used for display to end-users.

4.8.3 Viral metagenomics

Viruses also present additional set of taxonomy related challenges. Viruses constitute a large group of diverse entities that have thus far been largely uncharacterized and have defied a coherent unified taxonomic classification scheme. In addition, there are no good “markers” that define a virus (unlike bacteria where a number of housekeeping genes are universally found). Even composition-based or taxonomic methods may be insufficient as viruses frequently copy their genes from their host.

Viral diversity is estimated to be very high and viruses are important players in many environments (e.g. unknown viruses might be causing diseases in human). Currently there are many practical hurdles to their study and viral diversity is thought to be vastly under-represented in reference databases. It is not uncommon to identify long DNA segments in metagenomic mixtures that do not have good homology to any organisms, even distantly related. An example is the recent discovery of a new phage in HMP data (the crAss Phage [1]), phage that is found in about 50% of the human population yet its proteins only bear very distant similarities with other phage-like proteins. This problem is compounded by the fact that in many cases the abundance of a virus within a sample is very low (e.g., 1 out of every 20 million reads), making it impossible to estimate the parameters of statistical models for organism identification, or requiring substantial computational resources in order to process the large volumes of data necessary to ensure sufficient coverage. Different participants reported that in their experience, the targeted enrichment of viruses do not work well. RNA viruses pose further difficulties, for instance, in getting starting material for sequencing. There are iterative approaches to close in on the virus sequence using a combination of assembly and reference search per step. However, finding a virus in a sample does not always necessarily mean it is relevant for a given diseases phenotype.

Another opportunity for interesting research is creating host-virus linkages (phage-bacterium, defining host range for eukaryotic viruses, etc.).

4.8.4 Database issues

4.8.4.1 Correctness of reference databases

An additional challenge is posed by the databases themselves. In many cases the sequences deposited in public databases are mis-annotated, contain contaminants (e.g., mycoplasma in the human genome [2]), or represent “enriched metagenomes” rather than isolates (e.g., the sequences from this paper [3] which are deposited as isolates).

This state of affairs allows the opportunity for interesting research projects, such as the automatic identification of errors or contamination in public datasets. These could be phrased as outlier detection at different levels of resolution: – inconsistent taxonomic placement of an entire assembly – inconsistent taxonomic placement of individual contigs or genomic regions – outliers in terms of nucleotide composition, – discordant sequences not found in other genomes with the same label. A challenge in doing such analyses is avoiding the mis-classification of “true outliers”, such as ribosomal RNA operons or mobile elements that are genuine biological phenomena and not simply mis-annotations.

4.8.4.2 Databases of metagenomic datasets

There was also a discussion on publicly available resources for comprehensive collections of metagenome datasets. Resources include EBI, NCBI’s SRA, HMP DACC, CAMI, MetaHIT, iMicrobe, MG-RAST, etc. When analyzing new datasets, it is also important to be able to leverage existing datasets and inferences as much as possible. Frameworks and representation schemes that do not require expensive recomputations should be developed. On this front, challenges and opportunities include [a] the efficient representation of a comprehensive metagenome database (ideas from data representation for pan-genomics data could be applicable here), [b] index construction from large reference datasets, with an updateable index being highly desirable (for instance, allowing for easy addition and removal of strains depending on task), [c] adaptable alignment strategies to allow for more variability and recombinants in viral genomes (for instance, mapping against a pan genomic viral database); ability to deal with different use cases (viral with high variability, fungi with lower variability and different index), [e] mapping against pan-genome graphs, [f] approaches for build these pan-genome graphs (streaming/online concepts, succinct data structures), and [g] need for visualization and higher level access.

4.8.5 Quantitative metagenomics

While the metagenomic paradigm has been very useful in understanding the taxonomic and functional make-up of microbial communities, it is also important to understand the limitations of the framework used by most metagenomic studies. From data generated by whole genome shotgun sequencing, and in the absence of any calibration information related to quantitation (like spike-in of known amounts or qPCR data), the inferred taxonomic or functional profiles reflect relative abundances, and not absolute abundances, of taxonomic or functional groups. Thus it is not possible to assess microbial load (i.e. total number of microbes per unit volume or per unit mass) from these data, impacting our ability to answer important questions in many areas of microbiome research including for several biomedical applications.

The importance of data transformation of read count data was also discussed in the context of metagenome comparisons and identification of differentially abundant groups. Data transformations depend on the downstream analyses and include corrections for gene

copy number (e.g.: 16S gene copy numbers) and genome sizes. However the choice of the transformation function(s) is not always clear. Any comparison between metagenomic datasets also has to consider possible study specific biases associated with sample preparation and sequencing. To be able to compare metagenomic samples, it is important to define dissimilarity measures between samples. These measures may be based on taxonomic or functional profiles, or even derived directly from the underlying sequence similarities of raw reads. For many microbiome evaluations (like in the case of human microbiome comparisons), it is important to understand the distributions (of taxonomic and functional profiles) of reference population groups (like “healthy” individuals).

4.8.6 Functional annotation

Functional analysis of metagenomic data typically involves the annotation of predicted genes using resources like KEGG and MetaCyc/BioCyc that organize protein function at various levels including pathways. Analysis approaches then typically compute abundances of functional groups at these different levels. However, it is not obvious whether concepts like pathway abundances for metagenome data have any biological interpretation. Even the simpler computational challenge of pathway detection (in a given taxon in a metagenomic dataset) is complicated by the observation that, in pathway organization, genes are often assigned to multiple pathways, and these dependencies need to be explicitly modeled. Methods for increasing identification and resolution of functions of genes in metagenomic datasets were also discussed, including the use of co-localization information on the genome as a marker for membership in the same pathway or functional unit. Challenges in annotation of genes related to properties like Antimicrobial Resistance, Virulence, and Pathogenicity were also discussed. When annotating a gene in a metagenomic dataset for one of these properties, it is not always possible to make the inference based on a simple homology search against reference databases (like CARD, VFDB etc.). The conditions and constraints that lead to these properties are rather complex, and cannot be modeled by a simple criterion involving presence/absence of a particular gene. We need new approaches to express such constraints and conditions, and new ways to search against such complex rule sets.

4.8.7 Metagenomic assembly

Obtaining high quality genome assemblies from metagenomic datasets remains a challenge, with reconstruction of strains being a particularly challenging problem currently. Current binning tools are quite good for strain reconstruction in cases when there are not too many closely related strains. For real metagenomic data, however, misassembly is almost indistinguishable from novel strains occurring in species with high-recombination rates (read evidence is necessary to confirm the recombination event). Techniques for parsing strains from pangenome-based formalisms may be worth exploring in this context. Binning and assembly techniques based on co-occurrence patterns of contigs across multiple samples were also discussed.

4.8.8 Benchmarking and validation

Benchmarking of methods for metagenome analyses were also discussed. Common tasks to benchmark include assembly, taxonomic binning (bins represent sequences originating from one taxon; this corresponds to individual strains at the lowest level of the taxonomy), and taxonomic profiling (estimation of frequencies of different taxa in a sample). Benchmark dataset design criteria and evaluations are important components of any benchmarking. For

strain analyses, goals would be to reconstruct genomes of individual strains and to distinguish between closely related strains, representing different degrees of evolutionary relatedness in a data set: [a] given a mixture, identify species represented by an individual strain, [b] given a mixture, distinguish between two or more strains of the same species that are present simultaneously without recombination, and [c] simulate recombination within the sample.

Ways to generate benchmark datasets include [a] simulating mixing of sequences starting from isolate sequences, [b] creating mock communities by “mixing DNA together” and sequencing the mock community, and [c] using real data sets, and hold back reference genome sequences of community members as standard of truth that have been isolated and sequenced in addition from the same community.

Benchmark dataset generation should model different variables (sequencing technology, read length, coverage, insert size, etc.). Inclusion of plasmids and other kinds of non-bacterial material into the benchmark set would enable the creation of more realistic datasets.

References

- 1 Bas E. Dutilh, Noriko Cassman, Katelyn McNair, Savannah E. Sanchez, Genivaldo G. Z. Silva, Lance Boling, Jeremy J. Barr, Daan R. Speth, Victor Seguritan, Ramy K. Aziz, Ben Felts, Elizabeth A. Dinsdale, John L. Mokili and Robert A. Edwards. *A highly abundant bacteriophage discovered in the unknown sequences of human faecal metagenomes*. Nature Communications, 5, 4498, 2014
- 2 William B Langdon. *Mycoplasma contamination in the 1000 Genomes Project*. BioData Mining, 7:3, BioMed Central, 2014
- 3 Mette T Christiansen, Amanda C Brown, Samit Kundu, Helena J Tutill, Rachel Williams, Julianne R Brown, Jolyon Holdstock, Martin J Holland, Simon Stevenson, Jayshree Dave, CY William Tong, Katja Einer-Jensen, Daniel P Depledge and Judith Breuer. *Whole-genome enrichment and sequencing of Chlamydia trachomatis directly from clinical samples*. BMC Infectious Diseases, 14:591, BioMed Central, 2014

4.9 Haplotype Phasing

Knut Reinert (FU Berlin, DE), Niko Beerenwinkel (ETH Zürich – Basel, CH), Jason Chin (Pacific Biosciences – Menlo Park, US), Richard Durbin (Wellcome Trust Sanger Institute – Cambridge, GB), Mohammed El-Kebir (Brown University – Providence, US), Anne-Katrin Emde (New York Genome Center, US), Gunnar W. Klau (CWI – Amsterdam, NL), Veli Mäkinen (University of Helsinki, FI), Tobias Marschall (Universität des Saarlandes, DE), Alice Carolyn McHardy (Helmholtz Zentrum – Braunschweig, DE), Siavash Mirarab (University of California at San Diego, US), Kay Nieselt (Universität Tübingen, DE), Bernhard Renard (Robert Koch Institut – Berlin, DE), Enrico Siragusa (IBM TJ Watson Research Center – Yorktown Heights, US), Peter F. Stadler (Universität Leipzig, DE), Granger Sutton (The J. Craig Venter Institute – Rockville, US), Tandy Warnow (University of Illinois – Urbana-Champaign, US), and David Weese (SAP Innovation Center – Potsdam, DE)

License © Creative Commons BY 3.0 Unported license

© Knut Reinert, Niko Beerenwinkel, Jason Chin, Richard Durbin, Mohammed El-Kebir, Anne-Katrin Emde, Gunnar W. Klau, Veli Mäkinen, Tobias Marschall, Alice Carolyn McHardy, Siavash Mirarab, Kay Nieselt, Bernhard Renard, Enrico Siragusa, Peter F. Stadler, Granger Sutton, Tandy Warnow, and David Weese

4.9.1 Problem definition

Haplotype phasing describes the problem of reconstructing the individual haplotypes of a polyploid organism.

Different cases can be distinguished which alter the computational problem.

- Diploid genome
- Polyploid genome
- unknown ploidy (RNA-viruses, but also repeats in assembly, metagenomics clonotypes or strains)

4.9.2 Approaches

There are in general 3 different approaches to solve the problem:

- Read-based: The Next Generation Sequencing (NGS) reads obtained from sequencing machines can stem from any of the organisms genomic strands. From which is not known. Hence we have to infer an assignment of each read to a reconstructed haplotype. This can be done via the help of an MSA or without (alignment free). The problem is in general harder if the ploidity is unknown.
- Information from other experiments (arrays, etc.)
- Population approaches (trios (or available pedigree) or other groups of related individuals)

4.9.3 Discussion points

- Is it solved given that we have long reads (or will have cheap long reads some time in the future)? ⇒ No. It helps of course, but depends on SNP frequency, error rate, sequencing depth
- Depending on the problem (e.g. potato has high SNP frequency) various technologies can be applied (long or short reads)
- The problem can be simplified or solved using approaches from molecular biology (e.g. separating haplotypes by microfluidics, inbreeding)

- The problem is the complement problem to error correction (either its a sequencing error or a SNP)
- It is confounded by possible other error sources (sequencing errors, MSA errors)
- Ploidy > 4 cannot needs several SNPs to be resolved. Ploidy (from a computational perspective) not a constant number either globally or locally.

4.9.4 Challenges

- How to integrate other data (Hi-C), how to joint phasing (analysis of multiple samples)?
- Can scaffold information and haplotype information be integrated?
- How to estimate uncertainty (conflict between probabilistic methods and optimization)
- De novo / reference-free haplotyping (for bad or non existing reference genomes, see [1] that gives a partition of the reads which could help assembly)
- Simulators should be adapted to take into biological parameters into account
- Can visualization of phased haplotypes (on the population scale) [2] help for optimization of haplotypes?

References

- 1 Mikko Rautiainen, Leena Salmela and Veli Mäkinen. *Identification of Variant Compositions in Related Strains Without Reference*. Algorithms for Computational Biology, LNCS volume 9702, pp. 158-170, 2016, Springer Verlag
- 2 Günter Jäger, Alexander Peltzer and Kay Nieselt. *inPHAP: Interactive visualization of genotype and phased haplotype data*. BMC Bioinformatics, 2014, BioMed Central

4.10 Pan-Genomics

Knut Reinert (FU Berlin, DE), Jason Chin (Pacific Biosciences – Menlo Park, US), Fabio Cunial (MPI – Dresden, DE), Simon Gog (KIT – Karlsruher Institut für Technologie, DE), André Kahles (ETH Zürich, CH), Birte Kehr (deCode Genetics – Reykjavik, IS), Oliver Kohlbacher (Universität Tübingen, DE), Ben Langmead (Johns Hopkins University – Baltimore, US), Alice Carolyn McHardy (Helmholtz Zentrum – Braunschweig, DE), Siavash Mirarab (University of California at San Diego, US), Kay Nieselt (Universität Tübingen, DE), Enno Ohlebusch (Universität Ulm, DE), Adam M. Phillippy (National Institutes of Health – Rockville, US), Simon J. Puglisi (University of Helsinki, FI), Gunnar Rätsch (ETH Zürich, CH), Karin Remington (Computationality, US), Bernhard Renard (Robert Koch Institut – Berlin, DE), Peter F. Stadler (Universität Leipzig, DE), Granger Sutton (The J. Craig Venter Institute – Rockville, US), German Tischler (MPI – Dresden, DE), and Shibu Yooseph (University of Central Florida – Orlando, US)

License © Creative Commons BY 3.0 Unported license

© Knut Reinert, Jason Chin, Fabio Cunial, Simon Gog, André Kahles, Birte Kehr, Oliver Kohlbacher, Ben Langmead, Alice Carolyn McHardy, Siavash Mirarab, Kay Nieselt, Enno Ohlebusch, Adam M. Phillippy, Simon J. Puglisi, Gunnar Rätsch, Karin Remington, Bernhard Renard, Peter F. Stadler, Granger Sutton, German Tischler, and Shibu Yooseph

4.10.1 Topics

Interesting topics to be studied in this field:

1. Coordinates, data structure / graph
2. Annotation

3. Query support / questions
4. Tools /standards : transition of existing tools to the pan-genome
5. How to update an existing pan-genome with a new member
6. Taxonomic / evolutionary scale
7. Storage formats

4.10.2 Data structures

Use cases for coordinate system approaches:

1. E. coli's genomes are much more dynamic than for example human genomes
2. One approach computing an explicit coordinate system is the SuperGenome [1]: Based on a WGA, the SuperGenome is defined by the concatenation of all locally collinear blocks computed from the WGA. By this the coordinate system of the SuperGenome is derived from the alignment coordinates of all concatenated blocks. Furthermore, it defines an injective mapping of each individual genome into the global coordinate system defined by the SuperGenome.

Issues when working with a pan-genome defined by a global coordinate system is the ability to map between different coordinate systems and to update it (see below).

It was also suggested to define no coordinate system, but to construct the pan-genome just consisting of blocks. The question arises how to define the blocks.

Pan-genomes defined as a graph structure:

1. Though the pan-genome graph may be cyclic, no path of an individual genome in such a graph should contain a loop.
2. Should the graph be stored / transformed into a DAG? Because many operations on a DAG are much easier than on a cycle graph.
3. One version of storing the graph is storing all paths.
 - a. Even if we store the graph, what would be an efficient way to encode the set of all observed paths? Sparse bitvectors?
4. Provocative question: What could be the reason to store the graph structure? For visualisation for example. Adjacency of genes.
 - a. Why not a context-free grammar? There are efficient data structures for that.
5. What is different between different subsets of genomes within the pan-genome. What is common? What is proximal?
6. Transition probabilities on arcs?

4.10.3 Annotation

One issue is: given an annotation (in gtf format say), how is the annotation transferred to the pan-genome? Difficult to do on the graph, rather straightforward in a coordinate system

In the graph: annotations should be placed onto paths rather than on the whole graph by itself. Should the annotation also be stored together with the paths? Or independently stored? Or better to have a data structure that is able to quickly identify say conserved annotations. One other possibility is to define blocks by annotations.

4.10.4 Queries

Alignment queries:

1. Align a read to the pan-genome. Exact matching against a graph is possible, papers have also studied the efficiency of this process. Return:
 - a. just matches observed in the input sequences;
 - b. also combinations that are never observed in the input sequences.

Analysis/domain queries:

1. Gene finding: In the graph, for example identify “genes”, say, which in the graph would correspond to some kind of “bundles” (linear pieces) with a limited “thickness”.
2. Co-occurrences of alleles
3. Searching for certain graph structures
4. Use expression data for example to find genes
5. Find “motifs”.
6. GWAS-like comparisons: How does one group compare against another with respect to the graph structure?

String queries:

1. Compute the probability that a given short string S occurs in a genome described by the pan-genome.
 - a. What happens if S contains flexible, rigid gaps?
 - b. What happens if S is a PWM (e.g. as done during motif finding)?
2. Compute the probability of finding pattern T in a genome described by the pan-genome, given the occurrence of pattern S .
 - a. Probability that two patterns never occur together?
 - b. Probability that two patterns overlap?
3. Compute the probability of a new genome given a pan-genome (e.g. to decide whether a genome is more similar to one pan-genome than to another).
4. Given a short string S , return the set of all possible variants of S in a genome represented by the pan-genome. This allows to display e.g. all known variants of a gene or of a regulatory region simultaneously.
 - a. Compute the probability that a variant S' of S , rather than S itself, occurs in a genome.
5. Given a new genome S and a threshold k , return the k paths Q in the pan-genome with highest conditional probability $P(Q|S)$. Such paths could be used to input data that is missing from S , and to annotate S with a candidate recombination structure.
6. Given a new genome S , compute the average length of a recombination fragment, the probability of a recombination at a given position or inside a given interval, over all possible parses of S according to the pan-genome.
7. Given a threshold k , find a set V of k variation loci which maximize the number of variation loci not in V that can be predicted using V . This query, called tag selection, recurs in the design of SNP arrays.
8. Given a regulatory region in a genome represented by the pan-genome, and given a binding model for transcription factors, compute a probability distribution over all configurations of transcription factors bound to the regulatory region.
 - a. Compute a probability distribution over all possible translated sequences of a gene.
 - b. What is the probability that a given region in the pan-genome accomplishes a given function? (e.g. that it is an enhancer of gene expression)

9. Given a short string S , compute the set of all strings that can replace S and still produce valid genomes according to the pan-genome. Such “synonyms” might correspond e.g. to all possible variants of a given transcription factor binding site along a genome, which allow the same TF to regulate simultaneously multiple genes in different ways.
10. Given a short string S , let its context be the set of left- and right- extensions of S of a given length k . Compute the contingency table of two strings S and T , based on their sets of contexts in the pan-genome.
11. Assume that some paths in the pan-genome are marked. Compute some notion of “most discriminant features” between the marked and the unmarked paths.

Update queries to the data structure:

1. add a new unary path;
2. add a new genome;
3. “merge” N pan-genome data structures;
4. “concatenate” pan-genome structures created separately for distinct regions of the genome (e.g. for specific genes or linkage disequilibrium regions).
5. how frequent are updates in practice? and of which type?
6. One reason not to update is the change of the coordinate system.

Probably better to merge the different pan-genomes into one graph structure, because comparing different graphs tend to be very hard.

4.10.5 Tools / standards

1. Toolkit VG used on graph structures by the Durbin group (read alignment, construct the graph, ...). URL: <https://github.com/vgteam/vg>
2. Apart from that: how to devise a gene finder, and other tools on top of primary constructions
3. Comparing two pan-genomes, how to do that? Examples: two virus populations, two subpopulations of humans, two snapshots of a bacterium, cross compare tumors.

4.10.6 Taxonomic breadth

When does the pan-genomic concept break down? And when does it cease to help for e.g. the querying tasks?

References

- 1 A. Herbig, G. Jäger, F. Battke and K. Nieselt. *GenomeRing: alignment visualization based on SuperGenome coordinates*. *Bioinformatics*, 28:12, pp. i7–i15, Oxford Journals, 2012.

Participants

- Niko Beerenwinkel
ETH Zürich – Basel, CH
- Ewan Birney
European Bioinformatics
Institute – Cambridge, GB
- Christina Boucher
Colorado State University –
Fort Collins, US
- Jason Chin
Pacific Biosciences –
Menlo Park, US
- Pascal Costanza
Intel Corporation, BE
- Anthony J. Cox
Illumina – United Kingdom, GB
- Fabio Cunial
MPI – Dresden, DE
- Richard Durbin
Wellcome Trust Sanger Institute –
Cambridge, GB
- Mohammed El-Kebir
Brown Univ. – Providence, US
- Anne-Katrin Emde
New York Genome Center, US
- Simon Gog
KIT – Karlsruher Institut für
Technologie, DE
- Hannes Hauswedell
FU Berlin, DE
- Daniel H. Huson
Universität Tübingen, DE
- André Kahles
ETH Zürich, CH
- Birte Kehr
deCode Genetics – Reykjavik, IS
- Gunnar W. Klau
CWI – Amsterdam, NL
- Oliver Kohlbacher
Universität Tübingen, DE
- Ben Langmead
Johns Hopkins University –
Baltimore, US
- Pietro Lio'
University of Cambridge, GB
- Veli Mäkinen
University of Helsinki, FI
- Tobias Marschall
Universität des Saarlandes, DE
- Alice Carolyn McHardy
Helmholtz Zentrum –
Braunschweig, DE
- Siavash Mirarab
University of California at San
Diego, US
- Laurent Mouchard
University of Rouen, FR
- Gene Myers
MPI – Dresden, DE
- Luay Nakhleh
Rice University – Houston, US
- Kay Nieselt
Universität Tübingen, DE
- Enno Ohlebusch
Universität Ulm, DE
- Adam M. Phillippy
National Institutes of Health –
Rockville, US
- Mihai Pop
University of Maryland – College
Park, US
- Simon J. Puglisi
University of Helsinki, FI
- Gunnar Rätsch
ETH Zürich, CH
- Tobias Rausch
EMBL – Heidelberg, DE
- Knut Reinert
FU Berlin, DE
- Karin Remington
Computationality, US
- Bernhard Renard
Robert Koch Institut –
Berlin, DE
- S. Cenk Sahinalp
Simon Fraser University –
Burnaby, CA
- Enrico Siragusa
IBM TJ Watson Res. Center –
Yorktown Heights, US
- Peter F. Stadler
Universität Leipzig, DE
- Granger Sutton
The J. Craig Venter Institute –
Rockville, US
- German Tischler
MPI – Dresden, DE
- Esko Ukkonen
University of Helsinki, FI
- Tandy Warnow
University of Illinois –
Urbana-Champaign, US
- David Weese
SAP Innovation Center –
Potsdam, DE
- Shibu Yooseph
University of Central Florida –
Orlando, US

