EMBnet.journal

Volume 17 Nr. 1
July 2011

- SEQscoring: a tool to facilitate the interpretation of data generated with next generation sequencing technologies
- e-RGA: enhanced Reference Guided Assembly of Complex Genomes

and more...

Editorial

Welcome to a new issue of EMBnet.journal the international, Open Access, peer-reviewed bioinformatics journal.

In this issue, amongst other articles, you will read about SeqAhead, a new COST Action initiative that aims to tackle many of the complex informatics issues generated by the deluge of data flowing from Next Generation Sequencing (NGS) technologies.

Genome sequencing studies are showing promise of both diagnostic and therapeutic benefits in clinical settings; in the agricultural area, the complete genome sequences of important plants, like the tomato and the potato, are out in the public domain - such projects facilitate the creation of new plant varieties that are resistant to drought and hash conditions. Clearly, a true biotechnological revolution has arrived, promising tangible biomedical and agri-food impacts. The need to train more researchers in the bioinformatics techniques necessary to deliver those impacts is therefore imperative; projects like SeqAhead are poised to help make this happen.

EMBnet.journal plays an integral part in the new data-driven landscape of the life sciences, not only providing updates on the latest tools, databases and educational materials relevant to the genomic revolution, but also offering general bioinformatics information to a growing community of researchers, from disciplines ranging from medicine to plant and farm animal sciences.

EMBnet.journal Editorial Board

proteinspotlight

> ONE MONTH, ONE PROTEIN <

Protein Spotlight (ISSN 1424-4721) is a periodical electronic review from the SWISS-PROT group of the Swiss Institute of Bioinformatics (SIB). It is published on a monthly basis and consists of articles focused on particular proteins of interest. Each issue is available, free of charge, in HTML or PDF format at http://www.expasy.org/spotlight.

We provide the EMBnet community with a printed version of issue 128. Please let us know if you like this inclusion.

Cover picture: Crowned Cranes, *Balearica regulorum*, Masai Mara, Kenya, 2010. [© Martin Norling]

Contents

Editorial Board:

Mastering Data-Intensive Collaboration and Decision Making through a Cloud Infrastructure: the Dicode EU project

Nikos Karacapilidis
University of Patras & RA CTI, Greece

Collaboration and decision-making settings are often associated with huge, ever-increasing amounts of multiple types of data, obtained from diverse sources, which have a low signal-to-noise ratio for addressing the problem at hand. In many cases, the raw information is so overwhelming that stakeholders are often at a loss to know even where to begin to make sense of it. In addition, these data may vary in terms of subjectivity and importance, ranging from individual opinions and estimations to broadly accepted practices and indisputable measurements and scientific results.

Nowadays, big volumes of data can be effortlessly added to a database. The problems start when we want to consider and exploit the accumulated data, which may have been collected over a few weeks or months, and meaningfully analyse them towards making a decision. Admittedly, when things get complex, we need to identify, understand and exploit data patterns; we need to aggregate large volumes of data from multiple sources, and then mine it for insights that would never emerge from manual inspection or from analysis of any single data source.

Taking the above issues into account, the recently funded Dicode project aims to facilitate and augment collaboration and decision making in data-intensive and cognitively-complex settings. To do so, it will exploit and build on the most prominent high-performance computing paradigms and large-data-processing technologies (such as cloud computing, MapReduce, Apache Hadoop, Apache Mahout and column databases) to meaningfully search, analyse and aggregate data existing in diverse, extremely large and rapidly evolving sources. Services to be developed and integrated in the context of the Dicode project will be released under an open source licence.

Building on current advancements, the solution foreseen in the Dicode project will bring together the reasoning capabilities of both the machine and humans. It can be viewed as an innovative "workbench", incorporating and orchestrating a set of interoperable services that reduce to a manageable level the data-intensiveness and complexity overload at critical decision points, thus permitting stakeholders to be more productive, and to concentrate on creative and innovative activities.

The achievement of the Dicode project's goal will be validated through three use cases. These were chosen to test the transferability of Dicode solutions in different collaboration and decision-making settings, associated with diverse types of data and data sources, thus covering the full range of the foreseen solution's features and functionalities.

The Dicode project is funded by the European Union under FP7. It started on September 1st, 2010 and its duration is 36 months. The partners of the Dicode consortium are: Research Academic Computer Technology Institute, Greece (project coordinator); University of Leeds, UK; Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e.V., Germany; Universidad Politécnica de Madrid, Spain; neofonie Gmbh, Germany; Image Analysis Ltd, UK; Biomedical Research Foundation, Academy of Athens, Greece; and Publicis Frankfurt GmbH, Germany.

- Dicode on the Web: http://dicode-project.eu/
- Dicode on Twitter: http://twitter.com/DICODE_EU
- Dicode on Facebook: http://www.facebook.com/people/Dicode-Eu/100001390513581

Competing interest statement
None declared

INTRODUCTION INTO SYSTEMS BIOLOGY

Basics of Proteomics, Bioinformatics, Biostatistics & Integration of Data Generated by these fields

at

Biomedical Research Foundation, Academy of Athens

The course will give a presentation of the principles, experimental processes involved, output and respective informatics needs, of the most commonly used proteomics methodologies generating vast amount of data. How to manage, analyze and make sense of these data using state-of-the-art bioinformatics approaches, with emphasis on the user end to assist the person with data generation, collection and analysis.

Organising committee
Sophia Kossida
Marc Baumann
Teresa Attwood
Antonia Vlahou
Spiros Garbis

The number of participants is limited to a maximum of 12 students or post-docs.
FEBS Youth Travel Fellowships are available for students from Europe.

Applications for YTF grants must include:

- short CV
- outline of research activities
- motivation letter, including potential benefits for the applying student as well as the home laboratory

For more information, please visit www.bioacademy.gr/bioinformatics/courses/FEBS_2011

The Bioinformatics Roadshow

1-3 November 2011

Athens, Greece

Biomedical Research Foundation

Academy of Athens

TOPICS

Day 1: Ensemble: Genome databases for vertebrates and other eukaryotes.

Day 2: UniProt, InterPro (from the Proteomes) and Reactome (from the Pathways).

Day 3: PDBe: Macromolecular structure database, including PDB search tools.

Registration fee: 90 €

Organised by EMBL_EBI in collaboration with BRFAA.

For more information, please visit

www.bioacademy.gr/bioinformatics/courses/bioinfo_roadshow2011

EMBL-EBI SIB BRENDA European Patent Office

ISCB and EMBnet to collaborate on bioinformatics education and training

ISCB Press Release

SAN DIEGO, USA and UPPSALA, SWEDEN - 26 April 2011 - The International Society for Computational Biology (ISCB) and the European Molecular Biology Network (EMBnet) are pleased to announce a collaboration to provide education and training in the use of bioinformatics tools to members of our communities. Having just sponsored and organised a successful workshop introducing the EMBnet eBioKit at the ISCB Africa ASBCB Conference on Bioinformatics 2011 (www.iscb.org/iscbafrica2011/), that followed a similarly sponsored and organised workshop on sequence analysis using EMBOSS at the 2009 rendition of the same conference (www.iscb.org/iscbafrica2009/), both EMBnet Executive Board member Erik Bongcam-Rudloff and ISCB President Burkhard Rost enthusiastically embraced the idea of a formal collaboration that will ensure EMBnet training courses are incorporated into ISCB meetings whenever possible, particularly those in developing regions.

"EMBnet is a very strong member of ISCB's Affiliated Regional Network, and formalizing our collaboration to incorporate their training into ISCB conferences makes perfect sense," commented Prof. Dr. Rost. "They do this very well and our member community will undoubtedly benefit from their expertise," he continued. The next meeting that may be targeted for inclusion of an EMBnet training workshop is the recently announced InCoB/ISCB-Asia conference taking place in Kuala Lumpur, Malaysia, November 27-December 2, 2011. Following that, discussion has already begun on incorporating an EMBnet training session in the next ISCB-Latin America conference, to be held in Santiago, Chile, in March 2012.

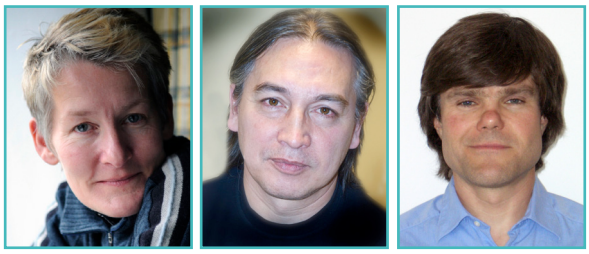
As EMBnet is comprised of 28 national and eight specialists nodes, strong ties to many of the active bioinformaticians in countries around the world are securely in place. Prof. Dr. Bongcam-Rudloff summed up the collaboration nicely: "This is a natural fit. ISCB is expanding its regional meetings into areas where EMBnet has close working relationships with high-level members of our network. Working together we will accomplish more, and everybody wins."

EMBnet has also worked closely with ISCB's Student Council, featuring their accomplishments in the EMBnet Journal, providing technical help with a virtual meeting organised by the students, and presenting an eBioKit tutorial to a joint meeting of several Regional Student Groups. Both Bongcam-Rudloff and Rost agree: "This collaboration establishes the framework for further involvement of EMBnet in ISCB Student Council activities. This is good for the future of our entire scientific community."

The International Society for Computational Biology (http://www.iscb.org) is the leading professional society for the new era of computational biology.
For further information contact info@iscb.org

EMBnet (http://www.embnet.org) is a science-based group of collaborating nodes throughout all continents.
For further information contact Erik.Bongcam@slu.se

SEQAHEAD - COST Action BM1006: Next Generation Sequencing Data Analysis Network



Teresa Attwood[1], Erik Bongcam-Rudloff[2], Andreas Gisel[3]

[1]University of Manchester, United Kingdom
[2]Department of Animal Breeding and Genetics, Swedish University of Agricultural Sciences and Department of Immunology, Genetics and Pathology, Uppsala University, Sweden
[3]CNR, Institute for Biomedical Technologies, Bari, Italy

COST (European Cooperation in Science and Technology) is one of the longest-running European instruments supporting cooperation, collaboration and orchestration among scientists and researchers across Europe working in the same field. Some of the organisers of the two EMBRACE workshops on 'Next Generation Sequencing' (NGS) saw this type of Action as exactly the right kind of mechanism to try to tame the data tsunami being generated by the furiously fast developing NGS technologies. Their aim was to tackle the bioinformatics challenges inherent in managing and analysing these data, and to support researchers who use NGS technologies but do not have direct access to the necessary underpinning bioinformatics resources. The history of the NGS initiative is short, but explosive. It is imperative for the life science community to be prepared for the enormous growth in NGS data, the challenges this presents, and the opportunities it affords. Recognising these issues, and the need for global cooperation, gave birth to the idea for this COST Action proposal; it developed into the concerted action of today.

The initial proposers of the Action, Erik Bongcam-Rudloff and Andreas Gisel, Eija Korpelainen and Peter Rice were members of Work Package 4 (WP4) Test Cases in the FP6 Network of Excellence (NoE) EMBRACE. Many of the test cases collected by WP4 involved problems relating to the use of NGS technologies,

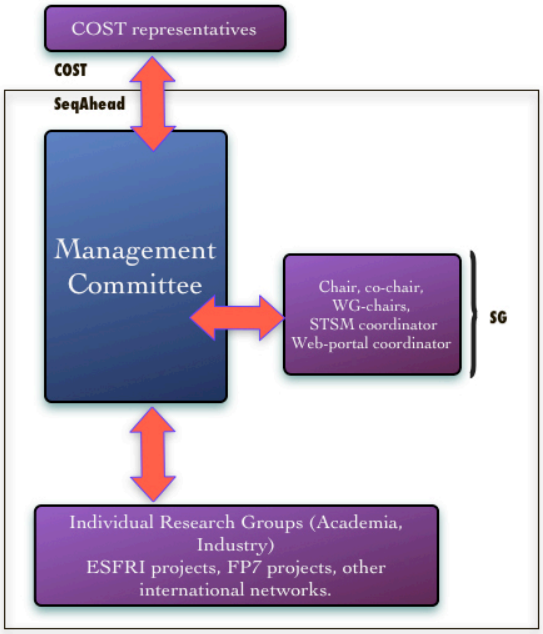


Figure 1. Graphic of the Action management structure.

demonstrating the growing need for NGS data analysis procedures of various types. As a first step to try to get to the bottom of the problems identified in WP4, the WP4 group was motivated to form a Task Force, including Erik Bongcam-Rudloff, Eija Korpelainen, Inge Jonassen, Nils-Einar Eriksson, Etienne deVilliers, Andreas Gisel, Laurent Falquet, JR Valverde and Gert Vriend. With the support of the Bioinformatics Italian Society (BITS), the Task Force organised an EMBRACE NGS data management and analysis workshop and hack-a-thon for bioinformaticians, which took place on November 18-20 (2009), in Rome at CASPUR, the Inter-University Consortium for the Application of Super-Computing for Universities and Research, hosted by Node manager Tiziana Castrignano. A comprehensive report on this workshop has been published on the EMBnet.news journal [1].

The acuteness of NGS data-analysis issues raised in the workshop led the Task Force to write a COST pre-proposal, *SEQAHEAD: Next Generation Sequencing Data Analysis Network*. The text proposed a strong network to monitor, on one side, the rapidly growing sequencing technologies and, on the other, the growing needs in data management and analysis. The idea was that a highly orchestrated action should guarantee efficient, agile procedures to cope with the NGS data flood. The pre-proposal

Figure 2. Group picture of the participants of the COST Action kick-off meeting in Brussels.

was written by six partners, in consultation with several of the workshop participants, who had indicated their interest in joining the Action if it were funded. In May 2010, the pre-proposal passed with almost the maximum score, and the way was therefore open to prepare a full proposal. In parallel with this development, and encouraged by the success of the first workshop, a second EMBRACE NGS workshop was organised on June 16 -17 (2010), in Ruvo di Puglia, Italy, affiliated with the EMBnet annual general meeting (for the programme, see [2]), with the full support of the Italian EMBnet Node manager Domenica D'Elia.

The latter meeting had multiple purposes. It was both a workshop to exchange ideas and information, and to work together, via a hack-a-thon, on real life NGS problems, and it was a platform to discuss the structure of, and ideas for, the full COST proposal.New partners among the participants were invited to sign up to the Action, and they drew our attention to colleagues not present at the meeting but who would likely also be interested to join (for the report, see [3]). In the end, forty-eight scientists with different knowledge within the NGS domain, from important institutions across sixteen European countries, created the full proposal. Again, this passed with almost the maximum score, and Erik Bongcam-Rudloff, the coordinator of the Action, successfully defended it in front of the Domain Committee. Early in December 2010, the proposed COST Action was officially approved [4].

With eighteen countries signed up to, or intending to sign, the Memorandum of Understanding (MoU), the COST Action BM1006 *Next Generation Sequencing Data Analysis Network* was initiated with a kick-off meeting on March 13 (2011), in Brussels. The meeting was organised by the COST office, and chaired by Magdalena Radwanska and Anja Van Der Snickt, who introduced the participants to the 'world' of COST Actions, mainly running us through the labyrinthine rules and regulations. They explained that the designated rapporteur for the Action will be Dr. Tanja Gmeiner Stopar (University Medical Centre Ljubljana, Slovenia), who will be responsible for monitoring our activities as the project moves forward. The primary goals of the kick-off meeting were to establish the budget for the first year, and to elect key personnel from within the consortium to manage the Action over the next four years. Erik Bongcam-Rudloff (SE) was consequently elected as the Action Chair and Teresa Attwood (UK) as Vice-Chair. A Chair and Vice-Chair were also allocated to each Working Group, as follows: WG1, *Technology watch for new developments*: Ralf Herwig (DE) and Thomas Svensson (SE); WG2, *Development of action plan for NGS bioinformatics to cope with challenges for the EU research area*: Andreas Gisel (IT) and Ana Conesa (ES); WG3, *Design, implementation and incorporation of software solutions*: Eija Korpelainen (FI) and Steve Pettifer (UK); WG4, *Generic informatics topics*: Veli Makinen (FI) and Alberto Policriti (IT); WG5, *Development of strategic dissemination and education program for NGS bioinformatics*: Gert Vriend (NL) and Jacques van Helden (BE). A number of other people were also elected to key roles within the Action: hence, Eric Rivals (FR) and JR Valverde (ES) will be responsible for coordinating the Short Term Scientific Missions (STSMs), which are mainly to support exchange activities of young researchers between European research institutes; and Sven Rahman (DE) will be responsible for the Action's Web portal [5], which will be an important medium for disseminating the Action's many activities in 'real-time'.

The budget for the first year was divided into four parts, mainly covering the first event, during October 2011, for all Management Committee members to meet, and to organise and start the Action's main activities. The rest of the budget will cover education, student exchange and publication and dissemination. Ultimately, the overall budget will be flexible, depending on the number of additional partners who sign up to the MoU as the Action progresses.

For now, we are extremely grateful to everyone who helped to get SEQAHEAD funded, and to those who have taken on important management roles within the Action; together, we eagerly look forward to vigorous and fruitful collaborations during the coming years!

References

1. http://journal.embnet.org/index.php/embnet-news/article/view/60/207
2. www.nextgenerationsequencing.org
3. http://journal.embnet.org/index.php/embnet-journal/article/view/176/382
4. http://w3.cost.eu/index.php?id=212&action_number=BM1006
5. http://www.seqahead.eu

Cutadapt removes adapter sequences from high-throughput sequencing reads

Marcel Martin
Department of Computer Science, TU Dortmund, Germany

Abstract
When small RNA is sequenced on current sequencing machines, the resulting reads are usually longer than the RNA, and therefore contain parts of the 3' adapter. That adapter must be found, and removed, error-tolerantly from each read before read mapping. Previous solutions are either hard to use or do not offer required features, in particular, support for colour-space data.
As an easy-to-use alternative, we developed the command-line tool cutadapt, which supports 454, Illumina and SOLiD (colour space) data, offers two adapter trimming algorithms, and has other useful features.
Cutadapt, including its MIT-licensed source code, is available for download at http://code.google.com/p/cutadapt/.

Introduction

The lengths of individual nucleotide sequences (*reads*) output by second-generation sequencing machines have reached 35, 50, 100 bps and more. When DNA or RNA molecules are sequenced that are shorter than this length, especially in small RNA sequencing experiments, the machine sequences into the adapter ligated to the 3' end of each molecule during library preparation. Consequently, the reads that are output contain the sequence of the molecule of interest and also the adapter sequence. Essential first tasks during analysis of such data are therefore to find the reads containing adapters and to remove the adapters where they occur. Only the relevant part of the read is passed on to further analysis. In some cases, finding adapters is a sign of contamination, and the reads containing them must be discarded entirely. For both tasks, we suggest the use of cutadapt, a user-friendly command-line program that supports a variety of file formats produced by second-generation sequencers. It especially supports colour-space data as produced by Applied Biosystems' SOLiD sequencer.

Other solutions for adapter trimming exist. Some software libraries, such as HTSeq by Simon Anders (http://www-huber.embl.de/users/anders/HTSeq/) and Biostrings (Pages et al., http://bioconductor.org/packages/release/bioc/html/Biostrings.html) offer the essential error-tolerant trimming routines, but HTSeq does not consider insertions and deletions, and both require the user to be able to write their own programs that use those routines. Some read-mapping tools, such as SOAP (version 1) by Li *et al.* [1], MAQ by Li *et al.* [2] and Novoalign (http://novocraft.com/) can also trim adapters, but this is only useful if the reads are to be mapped with the respective program. If one wants to use a different mapping tool without trimming capability, such as BWA [3], the solution is to use a stand-alone adapter trimmer, such as cutadapt. We are aware of two other tools of this type. Vectorstrip is part of the EMBOSS package [4]. It was originally developed to recognise and remove vector sequence contamination, which makes it slightly cumbersome to use in high-throughput sequencing experiments. Vectorstrip does not find partial adapter matches and does not support colour space. The program fastx_clipper, part of the FASTX-Toolkit by Assaf Gordon (http://hannonlab.cshl.edu/fastx_toolkit/) is another command-line tool in the same spirit as cutadapt, but it also does not support colour space data.

Cutadapt is the only stand-alone tool that can correctly trim colour-space reads. It supports FASTQ, FASTA and also SOLiD .csfasta/.qual input files. It outputs results in FASTA or FASTQ format; gzip-compression of input or output files is automatically detected.

Implementation

Cutadapt is mainly written in Python, but for speed, the alignment algorithm is implemented in C as a Python extension module. The program was developed on Ubuntu Linux, but tested also on Windows and Mac OS X. It should also work on other platforms for which Python is available.

Features

The program was initially developed to trim 454 sequencing data collected by Zeschnigk *et al.* [5]. As insertions and deletions within homopolymer runs are common in 454 data, cutadapt

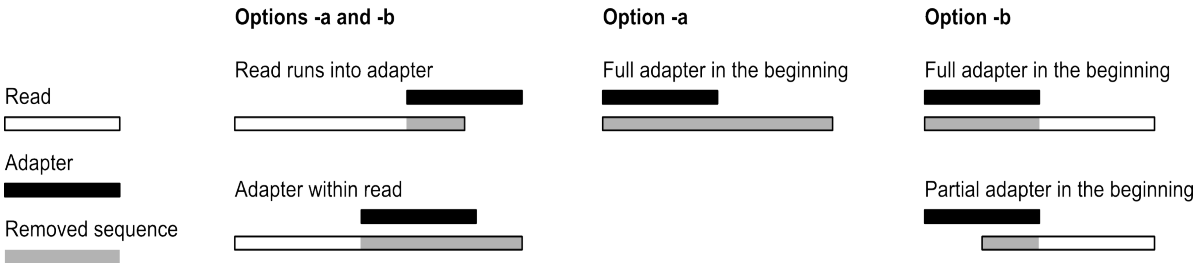


Figure 1. This illustration shows all possible alignment configurations between the read and adapter sequence. There are two different trimming behaviours, triggered by whether option "-a" or "-b" is used to provide the adapter sequence. Note that the case "Partial adapter in the beginning" is not possible with option "-a", as the alignment algorithm prevents it.

supports gapped alignment. For small RNA data analysis by Schulte *et al*. [6], the program was modified to support trimming of colour-space reads. It has also been tested and works well on Illumina data.

Cutadapt can search for multiple adapters in a single run of the program and will remove the best matching one. It can optionally search and remove an adapter multiple times, which is useful when (perhaps accidentally) library preparation has led to an adapter being appended multiple times. It can either trim or discard reads in which an adapter occurs. Reads that are outside a specified length range after trimming can also be discarded.

To decrease the number of random hits, a minimum overlap between the read and adapter can be specified. In addition to adapter trimming, low-quality ends of reads can be trimmed using the same algorithm as BWA. Cutadapt is thoroughly unit-tested. The program is actively maintained, and many features have been added in response to requests by users.

Performance

In theory, adapter trimming with cutadapt is dominated by the time needed to compute alignments, which is O(*nk*), where *n* is the total number of the characters in all reads, and *k* is the sum of the length of the adapters. In practice, other operations, such as reading and parsing the input files, take up more than half of the time. With 35 bp colour-space reads and an adapter of length 18, cutadapt trims approximately 1 million reads per minute (0.06 ms per read) on a single core of a 2.66 GHz Intel Core 2 processor.

Colour-space reads

Cutadapt correctly deals with reads given in SOLiD colour space. When an adapter is found, the adapter and the colour preceding it must be removed, as that colour encodes the transition from the small RNA into the adapter sequence and could otherwise lead to a spurious mismatch during read mapping. Cutadapt can produce output compatible with MAQ [2] and BWA [3]. These tools need FASTQ files in which the colours are not encoded by the digits 0-3, but by the letters ACGT (so-called double encoding), and in which the last primer base (given in nucleotide space) and the first colour are removed.

Usage example

In this simple example, the adapter AACCGG is trimmed from reads in the compressed file in.fastq.gz. The result is written to out.fastq:

```
cutadapt -a AACCGG in.fastq.gz > out.
fastq
```

Other use cases are documented in the README file in the cutadapt distribution, and on the website. Full documentation is available by typing "`cutadapt --help`" on the command line.

Algorithm

In the following, a character is a nucleotide or a colour (encoded by 0-3). The first step in processing a single read is to compute optimal alignments between the read and all given adapters.

Cutadapt computes either 'regular' or slightly modified semi-global alignments. Regular semi-global alignments, also called end-space free alignments [7, Chapter 11.6.4], do not penalise initial or trailing gaps. This allows the two sequences to shift freely relative to each other. When the "-a" parameter is used to provide the sequence of an adapter, the adapter is assumed to be ligated to the 3' end of the molecule, and the behaviour of

cutadapt is therefore to remove the adapter and all characters after it. With regular semi-global alignments, a short, usually random, match that overlaps the beginning of a read would lead to the removal of the entire read. We therefore require that an adapter starts at the beginning or within the read. This is achieved by penalising initial gaps in the read sequence, which is the only modification to regular overlap alignment. See Figure 1, right column.

Regular semi-global alignment is used when the location of the adapter is unknown (assumed when the "-b" parameter is used). Then, if the adapter is found to overlap the beginning of the read, all characters *before* the first non-adapter character are removed. See Figure 1 for an illustration of all possible cases.

After aligning all adapters to the read, the alignment with the greatest number of characters that match between read and adapter is considered to be the best one. Next, the error rate e/l is computed, where e is the number of errors, and l is the length of the matching segment between read and adapter. Finally, if the error rate is below the allowed maximum, the read is trimmed as shown in Figure 1.

Conclusion

Cutadapt solves a small, but important task within sequencing pipelines, especially those for small RNA. It offers an easy-to-use command-line interface. If colour space is to be processed, then cutadapt is the only standalone tool that supports this.

Acknowledgements

Part of this work was funded by Mercator Research Center Ruhr, Germany, grant Pr-2010-0016.

Competing interest statement

None declared

Bibliography

1. Li R, Li Y, Kristiansen K, Wang J (2008) SOAP: short oligonucleotide alignment program. Bioinformatics, 24(5), 713–714.
2. Li H, Ruan J, Durbin R (2008) Mapping short DNA sequencing reads and calling variants using mapping quality scores. Genome Res., 18(11), 1851–1858.
3. Li H, Durbin R (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics, 25(14), 1754–1760.
4. Rice P, Longden I, Bleasby A (2000) EMBOSS: the European Molecular Biology Open Software Suite. Trends Genet., 16(6), 276–277.
5. Zeschnigk M, Martin M, Betzl G, Kalbe A, Sirsch C, Buiting K, Gross S, Fritzilas E, Frey B, Rahmann S, Horsthemke B (2009). Massive parallel bisulfite sequencing of CG-rich DNA fragments reveals that methylation of many X-chromosomal CpG islands in female blood DNA is incomplete. Hum. Mol. Genet., 18(8), 1439–1448.
6. Schulte JH, Marschall T, Martin M, Rosenstiel P, Mestdagh P, Schlierf S, Thor T, Vandesompele J, Eggert A, Schreiber S, Rahmann S, Schramm A (2010) Deep sequencing reveals differential expression of microRNAs in favorable versus unfavorable neuroblastoma. Nucleic Acids Res., 38(17), 5919–5928.
7. Gusfield D (1997) Algorithms on Strings, Trees and Sequences. Cambridge University Press.

Command line analysis of ChIP-seq results

Endre Barta[1,2]

[1]Department of Biochemistry and Molecular Biology and Apoptosis and Genomics Research Group of the Hungarian Academy of Sciences, Hungary;
[2]University of Debrecen, Medical and Health Science Center, Research Center for Molecular Medicine, Egyetem ter 1. Debrecen, H-4010, Hungary

Abstract

Among the emerging next-generation sequencing technologies, ChIP-seq provides a very important tool for functional genomics studies. From the bioinformatics point of view, ChIP-seq analysis involves more than simply aligning the short reads to the reference genome. It also completes several other downstream steps, like determination of peaks, motif finding and gene ontology enrichment calculation. For these, several programs, applications and packages are available, both free and commercial. In this article I describe the usage of two free ChIP-seq analysis packages, the HOMER and ChIPseeqer along with the MACS and MEME programs. I also provide a customisable script suitable for the complete analysis of raw ChIP-seq sequencing data either from a sequence read repository or directly from sequencing.

Introduction

In the post-genomic era, ChIP-seq (Chromatin immunoprecipitation followed by next-generation sequencing) [1] soon became one of the most exciting technologies for functional genomic studies. Using ChiP-seq one can nearly determine the exact transcription factor binding sites (TFBSs) and histone modifications genome-wide. The experimental part of ChIP-seq consists of crosslinking proteins sitting on the chromosomes into the DNA, followed by a fragmentation step, which ideally yields 100-200 basepairs DNA-protein pieces. The next step is the immunoprecipitation of the specific fragments with a corresponding antibody. After some purification steps and library preparation, a short tag from either of the ends of the precipitated fragments will be sequenced using a next-generation sequencing method. The bioinformatic part of this analysis consists of the following three main steps:

- alignment of the short reads to the reference genome;
- finding significant peaks;
- downstream analysis, including de novo and known motif finding or analysis of the gene lists associated with the peaks.

In a typical ChIP-seq experiment it may be enough as less as 10 million reads for the saturation in an analysis. Hence, the main challenge today is neither the alignment of the reads to the reference genome, nor the peak finding, but rather the downstream bioinformatic analysis and the overall handling and maintaining of the different types of data generated by the analysis.

We would need to carry out a complex ChIP-seq analysis for the following reasons:

- to process our own experimental ChIP-seq data;
- to re-process already published ChIP-seq data to compare more detailed results with what the authors provided in the original articles;
- to make a meta-analysis of similarly processed ChIP-seq experiments from different sources.

In this article I describe an approach how it is possible to process ChIP-seq data from different experiments automatically, starting either from the SRA format files from NCBI [2], or FASTQ format files, or BAM format files which contain aligned reads. Among the many different available genome alignment tools I use the BWA. After aligning the short reads to the reference genome, the resulted SAM format files are converted on fly into the binary BAM format using the SAMtools program [3]. From the BAM format alignment files I use three different methods to find and analyse peaks. The first is the MACS (Model-based Analysis for ChIP-Seq) program [4] followed by the de novo motif finding using the MEME (Motif-based sequence analysis tools) program [5]. The second is the HOMER software [6] for motif discovery and ChIP-Seq analysis. The third one is the ChIPseeqer [7], which is a comprehensive framework for the analysis of ChIP-seq data. Using the three different approaches in parallel ensures that the analysis will be comprehensive and will cover every aspect of the possible questions.

Installation of the programs

Hardware requirements

To carry out such a comprehensive analysis we need a UNIX based computer. The operating system in principle can be any UNIX distribution. I have tested several LINUX and MacOSX Snow leopard based machines. Many steps in the analysis do not require any extra computing power, but the short read alignment, the peak and the *de novo* motif finding can run for a long time on slow processors. Furthermore, these steps need more memory and of course, the more memory we have the faster the finishing of the processes is. The storage capacity is an important factor as well. We need to have reference genome sequences and indices in place for the mapping. The raw sequencing reads and other files created during the analysis also need a lot of disk space. However, ChIP-seq analysis needs much less computing power, memory and storage capacity than other NGS tools. In summary, a PC with 1-2 terabytes (TB) of disk, minimum of 8 gigabytes (GB) of RAM and one recent processor with at least two cores can be enough, although in this case the alignment and the de novo motif finding steps can take days or even more than a week. I tested the programs on a mid 2010 Apple iMac computer (2.93 GHz Intel core i7, 16GB RAM, 2 TB disk), and it worked smoothly. Ideally, we need a machine with at least 16 GB of RAM, two processors with several cores, and at least 4 TB of raid storage.

Software environment

For the analysis several UNIX based (C, C++, PERL, PYTHON etc.) programs and scripts need to be installed. To run and compile these tools, beside the standard UNIX packages (like PERL, PYTHON or the wget), we will also need the developmental packages (Xcode for MacOSX or dev packages for LINUX distributions) for compiling programs from source code. For running MEME in parallel we need one of the MPI (Message Passing Interface) implementation and PBS (Product Breakdown Structure), for example if we want to run it on a supercomputing environment.

In general, installing the main C, C++ developmental packages, with their dependencies on a LINUX based machine, or the XCode package on a MacOSX based machine, can be enough. Otherwise, during the installation we can learn from the logs and error messages which package is missing or needs to be upgraded.

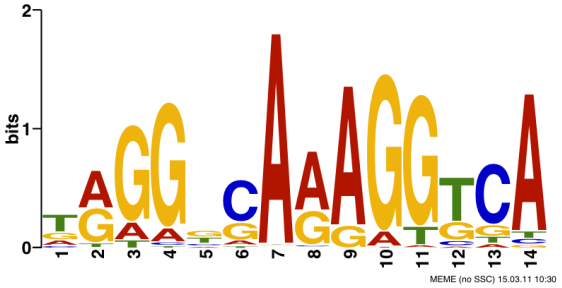


Figure 1. Motif finding result using parallel MEME on a macrophage PPARg ChIP result [14].

Specific programs/packages for the analysis

SRA toolkit is needed if we would like to download and process published raw ChIP-seq sequencing data from the NCBI SRA database. We can use both SRA and SRA-lite format data. Compiled binaries for several operating systems can be downloaded. After unzipping and untarring the files we only need to put the directory into the $PATH variable in the SHELL. The advantage of using the NCBI's SRA approach is that we need to transfer smaller files and we do not need to take care of the sequencing methods used.

BWA (Burrows-Wheeler Aligner) [8] is used for the alignment of the short reads to the reference genome. We need to download and compile the latest source code and to put it in the $PATH (I am using /usr/local/molbio/bin). For the alignment we also need to download and index the reference genomes. For this, I am using the human hg18 (because this is available for ChIPseeqer) and the mouse mm9 genome sequences. A script for making the indexing is available as well.

MACS is the most wildly used peak finding program. It is written in PYTHON, so we need to have it installed before installing MACS.

MEME is a de novo motif finding program, suitable for scan ChIP-seq peaks, or peak-centred regions for possible motifs (binding sites for the transcription factor used in the experiment). The main advantage of the MEME program is that it is still actively developed and it can be used under supercomputing environment. Compiling MEME on a grid computer needs properly installed MPI environment.

HOMER is a software package for motif discovery and ChIP-Seq analysis. Installing HOMER is very simple, we only need to have the proper

Information for motif1

Reverse Opposite:

p-value:	4.941e-324
log p-value:	-7.441e+02
Total Number of Sequences:	49999.0
Total Number of Target Sequences:	2876.0
Total Instances of Motif:	2072.4
Total Instances of Motif in Targets:	685.0
Motif File:	file (matrix) reverse opposite
PDF Format Logos:	forward logo reverse opposite

Figure 2. Motif finding result from the HOMER analysis on a macrophage PPARg ChIP result [14].

software environment and to download and run the *configureHomer.pl* script. There are also some third-party software needed for proper working of HOMER programs and scripts. The *configureHomer.pl* script is also suitable to download the different genomes for the analysis. In the HOMER web page there is a detailed instruction how to install and use the package.

ChIPseeqer [7] is an integrative and comprehensive framework that allows the users to perform in-depth analysis of ChIP-seq datasets through easily customised workflows. Besides the command-line tools, the newest ChIPseeqer also contains a GUI (Graphical User Interface) suitable for detailed analysis of ChIP-seq results. The ChIPseeqer package relies on some other programs (like FIRE, PAGE etc.) developed on the same laboratory [9]. In the latest version these programs are compiled together with the main programs and scripts, but we still need to manually set some variables (CHIPSEEQERDIR, FIREDIR, PAGEDIR, MYSCANACEDIR) and the PATHs for programs and PERL libraries.

For this analysis I have used the BWA version 0.5.9, MACS version 1.4.0rc2, which is working with PYTHON 2.6 to 2.7 (2.6.5 recommended). MEME version was 4.5 (4.6 available now with specific option for ChIP-seq region analysis). The HOMER version was 2.6 and the ChIPseeqer version was 1.0.

For converting the SAM files to BAM format I have used the SAMtools program, while for converting the BAM files to BED format for HOMER, I have used the BEDTools utility [10]. For tag statistics I have used the BAMTools utility [15].

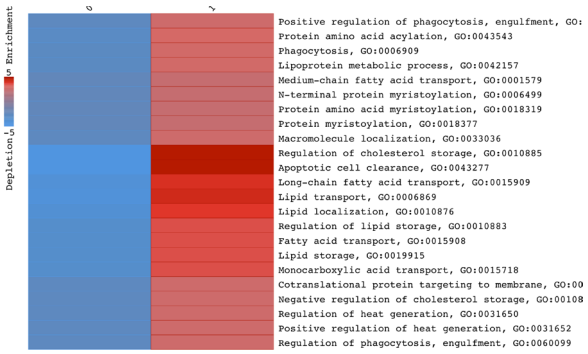


Figure 3. ChIPseeqer gene ontology analysis result on a macrophage PPARg ChIP result [14].

Implementation

The main goal for using these tools together is to generate data for comparison of the ChIP-seq results. Furthermore, this approach is suitable to process data from different sources. To make the analysis I had written two BASH SHELL scripts. The first is to run the programs and to carry out the analysis, while the second is to extract some statistics from the result. In the current implementation (version 1.1), the analysis script accepts either SRA or FASTQ (not colour spaced) format raw sequencing data, or BAM format alignments. If the BAM format alignment files are present in the proper location, the script omits the alignment steps. The script basically needs two parameters, the name of the file where the experiments to be analysed are listed, and the location of the basic directory for the analysis. It is also possible to provide an additional directory name, where the big files (SRA, FASTQ, SAI) will be stored and can be removed later safely. In the list file, there can be two fields separated by a space. The first is the name of the experiment starting with mm (mouse) or hs (human). I am using the format *hs _ celltype _ antibody _ condition*. The second field can be the FTP locations of SRA or SRA-LITE format raw sequencing data at the NCBI, or the FTP locations of fastq.gz format raw sequence data at the EBI SRA FTP site. If either the FASTQ or the BAM format file for the given experiment is available at the proper location with the proper name, e.g. *base _ directory/experiment _ name/bam/experiment _ name.bam*, this field can be

Table 1. Example of the statistics extracted from the log and result files of the analysis using the get _ tag _ statistics-v1 _ 0. sh script. The raw sequence data for the basic analysis with the *ChIP-seq _ anal-v1 _ 0.sh* script were obtained from different sources [11-14].

Article	experiment	FASTQ (raw reads)		BAM (BWA mapping)		HOMER analysis								MACS analysis					ChIPseeqer analysis		
		No of reads	% unique reads	No of total reads	% reads mapped	total tags after filtering	Average tags per peaks	Peaks width	Frag-ment length	No of peaks	IP effi-ciency	% peaks filtered by local signal	% clonal peaks filtered	tag-size	Total tags	% tags filtered out	Frag-ment length	No of peaks	No of peaks	Avg peak height	Avg peak size
Mikkelsen	mm_L1_PPARg1_d7	17,885,244	69.8%	17,996,284	46.7%	7,641,053	1.0775	285	53	11,160	3.12	15.1%	0.01%	61	8,396,770	15.6%	243	8,561	1,558	27	572
	mm_L1_PPARg2_d7	11,938,762	63.8%	12,212,356	52.6%	5,695,805	1.0868	96	53	4,375	1.35	26.0%	0.00%	71	6,426,779	18.5%	301	3,793	818	26	669
	mm_L1_CTCF_d7	24,082,065	56.6%	24,527,548	62.8%	9,638,738	1.0505	240	111	54,437	31.17	8.3%	0.01%	36	15,405,658	19.3%	96	59,551	26,093	48	434
	hs_ASC_CTCF_d9	21,972,388	63.7%	22,315,270	47.3%	8,979,919	1.0086	280	173	48,311	40.04	5.2%	0.00%	76	10,546,650	15.6%	154	47,179	30,515	53	429
	Hs_ASC_PPARg1_d9	34,635,364	66.1%	34,775,194	30.3%	9,315,493	1.0043	255	136	48,327	11.54	10.3%	0.00%	76	10,534,909	11.9%	156	39,902	9,930	27	325
	Hs_ASC_PPARg2_d9	42,987,395	65.7%	43,123,317	16.4%	6,316,958	1.0031	244	133	31,581	8.89	5.7%	0.00%	76	7,062,963	10.8%	154	27,245	5,749	23	301
O'Geen	hs_GM_TR4_2	28,598,265	83.9%	28,177,320	79.5%	21,896,016	1.0056	264	41	10,387	2	30.5%	0.13%	32	22,400,007	1.3%	33	7,968	1,616	36	402
	hs_HeLa_ELK1_1	69,333,573	36.3%	43,345,862	54.2%	21,241,382	1.0066	309	41	6,551	1.69	41.3%	0.12%	32	23,474,148	2.2%	33	6,581	1,587	30	332
	hs_HeLa_ELK4_2	51,888,861	23.0%	28,402,913	59.8%	14,107,350	1.0078	331	41	9,831	2.6	23.7%	0.09%	32	16,982,796	3.9%	34	11,898	2,170	25	315
	hs_HeLa_TR4_2	25,624,681	88.8%	25,129,235	52.6%	12,892,737	1.0033	155	106	9,926	2.38	18.3%	0.06%	32	13,208,389	0.9%	47	6,349	2,011	47	406
	hs_HepG2_TR4_2	16,347,187	85.7%	16,276,605	79.4%	12,509,867	1.0042	275	41	4,904	2	31.3%	0.17%	32	12,917,727	1.0%	35	4,671	1,307	39	430
	hs_K562_TR4_2	14,851,399	88.0%	14,855,527	72.1%	10,490,292	1.0057	192	41	2,059	1.28	60.1%	0.14%	32	10,705,127	2.6%	33	1,559	679	28	394
Nielsen	mm_L1_PPARg_d0	12,025,045	74.9%	12,025,037	74.1%	7,872,656	1.0375	52	85	6,502	1.38	21.5%	0.19%	32	8,911,122	14.9%	39	4,036	635	25	421
	mm_L1_PPARg_d1	13,588,664	63.1%	13,588,660	74.4%	8,325,879	1.0363	58	90	7,639	1.48	19.9%	0.43%	32	10,104,943	20.6%	40	4,226	731	26	463
	mm_L1_PPARg_d2	18,154,176	61.3%	18,154,159	40.4%	5,990,659	1.0482	96	41	15,961	3.8	7.6%	3.40%	32	7,339,281	22.1%	42	11,075	918	23	302
	mm_L1_PPARg_d3	14,392,918	71.8%	14,392,892	76.6%	9,615,931	1.0393	67	87	10,816	1.73	16.8%	0.19%	32	11,023,487	16.2%	40	5,191	721	29	632
	mm_L1_PPARg_d4	15,034,953	57.7%	15,034,953	60.2%	6,938,366	1.0419	70	80	19,773	3.34	8.4%	0.26%	32	9,048,496	10.4%	50	13,936	1,512	23	328
	mm_L1_PPARg_d6	15,140,624	62.1%	15,140,621	50.0%	6,094,142	1.0392	94	88	36,277	8.67	5.8%	0.32%	32	7,573,685	22.6%	69	34,417	6,340	27	330
	mm_L1_RNAPII_d0	8,573,748	82.0%	8,573,748	89.9%	7,204,654	1.0208	100	106	22,922	5.45	25.2%	0.00%	32	7,707,149	8.4%	111	15,948	4,665	23	368
	mm_L1_RNAPII_d1	9,715,568	80.9%	9,715,568	83.7%	7,500,956	1.0181	89	78	32,578	8.07	39.8%	0.01%	32	8,127,267	9.3%	70	19,524	7,137	25	422
	mm_L1_RNAPII_d2	10,318,412	73.4%	10,318,404	80.7%	7,308,686	1.0173	90	88	32,151	8.06	42.0%	0.01%	32	8,324,284	13.7%	85	18,056	6,803	24	430
	mm_L1_RNAPII_d3	8,905,079	81.4%	8,905,071	83.7%	6,890,426	1.0155	84	86	30,990	7.4	36.3%	0.01%	32	7,453,351	9.0%	86	18,140	5,588	23	428
	mm_L1_RNAPII_d6	8,979,276	65.1%	8,979,275	87.4%	6,502,922	1.048	48	79	12,327	2.44	23.1%	0.01%	32	7,850,700	21.0%	70	10,704	1,293	21	381
	mm_L1_RXR_d0	7,767,412	73.0%	7,767,407	67.7%	4,634,488	1.0153	61	100	17,134	3.78	7.0%	0.30%	32	5,259,011	13.2%	49	10,298	928	20	285
	mm_L1_RXR_d1	8,549,694	67.8%	8,549,651	81.3%	5,923,120	1.0182	82	96	32,111	5.74	6.6%	0.05%	32	6,953,474	16.3%	93	22,644	2,844	21	302
	mm_L1_RXR_d3	9,503,443	68.6%	9,503,443	63.4%	5,093,136	1.0317	71	78	38,857	6.22	4.5%	0.10%	32	6,021,891	18.0%	56	13,198	1,619	21	294
	mm_L1_RXR_d4	14,468,617	44.9%	14,468,610	40.8%	3,948,339	1.0334	73	85	36,722	8.49	4.3%	0.47%	32	5,902,838	35.3%	52	17,633	2,302	20	286
	mm_L1_RXR_d6	9,395,192	68.0%	9,395,188	66.1%	5,191,748	1.0273	93	92	39,302	9.95	5.5%	0.11%	32	6,207,267	18.6%	87	32,595	6,290	25	321
Lefte-rova	mm_L1_PPARg	7,502,114	66.0%	7,502,114	88.9%	5,660,584	1.0483	102	41	5,540	1.82	19.3%	0.06%	36	6,668,116	19.0%	79	6,866	618	27	863
	mm_mac_CEBP	7,898,957	42.6%	7,898,957	84.7%	4,665,715	1.0335	98	81	38,024	13.44	3.8%	0.01%	36	6,689,884	32.5%	80	34,765	10,310	26	307
	mm_mac_PPARg	17,389,919	66.9%	17,323,524	75.1%	11,348,348	1.0338	121	41	3,769	0.81	20.9%	0.29%	36	13,002,396	6.9%	39	2,635	500	30	386
	mm_mac_PU1	9,438,501	65.9%	9,382,873	77.0%	6,278,059	1.0155	105	82	60,996	17.75	4.0%	0.00%	36	7,220,167	14.4%	76	48,032	15,728	26	300

empty. The script will create the directory structure for the experiments and run the programs. All the logs and standard error messages related to program running will be put to the *base _ directory/experiment _ name/logs* directory, while the results will be put to the *bam, macs, homer and chipseeqer* directories.

The scripts performs the most basic ChIP-seq related analyses including the alignment to the reference genome, peak finding using MACS, HOMER and ChIPseeqer, quality control calculations using HOMER and ChIPseeqer, *de novo* motif finding using MEME (Fig. 1), HOMER using FindPeaks (Fig. 2) and ChIPseeqer. HOMER and ChIPseeqer also carry out a detailed annotation of the peak regions as well as a Gene Ontology analysis (Fig. 3). They also have some other programs and scripts suitable for many different tasks connected to the analysis. These include, among others, the comparing of the ChIP regions (peaks) and combining them into common sets. The results are mostly available as BED format files for peak regions that are suitable for visualisation in any of the available genome browsers. The results of de novo motif finding programs are available as local HTML pages (HOMER, MEME), or in EPS and PDF format files (ChIPseeqer). The annotation files generally can be found as tab delimited text files available for direct import into spreadsheet programs. The summary of statistics about the experiment is generated as comma separated values (CSV).

Conclusions

It is now getting easier and easier to carry out parallel ChIP-seq experiments using multiplexed next-generation sequencing. The SRA at the NCBI, and the ENA (European Nucleotide Archive) at the EBI, host more and more raw and processed ChIP-seq sequencing result. As a consequence, this growing data are available for detailed analysis and for comparing to our own results. To deal with this ever-increasing amount of next-generation sequencing data, we need bioinformaticians capable to work on a UNIX environment. For them, using command line tools for processing and comparing ChIP-seq data can be a good alternative to commercially available GUI version program packages. Here, I provide a layout with example scripts to conveniently analyse, and thus easily compare, ChIP-seq experiments from different sources. The method can be very useful

not only for processing our own data but also to compare our data to other's, or simply to make a ChIP-seq meta-analysis using the available ChIP-seq raw sequence data at the sequence read repositories. The main advantages of this approach are that: i) it can be run on a minimal, low-budget hardware; ii) provides comparable data for every aspect of the analysis; iii) is easy to customise for any personal needs.

Availability

The scripts are available upon E-mail request or from the Facebook Page "Command line ChIP-seq analysis". The Facebook Page is also meant to be a place for providing further details about installing and using these programs and scripts, for announcing improvements or new versions of programs and scripts, and also to exchange experiences about using command line tools for ChIP-seq analysis.

Acknowledgments

I would like to thank Prof. László Nagy and Bálint L. Bálint for inspiring and helping my work, and for the support of the Hungarian Academy of Sciences Bolyai Scholarships. This work was supported by grants from the Hungarian Science Research Fund OTKA NK72730, and the NKTH-ANR TéT BOV-rSNP. Some of the programs were running on the GenaGrid supercomputer, kindly provided by the GenaGrid consortium.

Competing interest statement

None declared

References

1. Park PJ (2009) ChIP-seq: advantages and challenges of a maturing technology. Nat Rev Genet 10: 669-680.
2. Leinonen R, Sugawara H, Shumway M (2011) The sequence read archive. Nucleic Acids Res 39: D19-21.
3. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, et al. (2009) The Sequence Alignment/Map format and SAMtools. Bioinformatics 25: 2078-2079.
4. Zhang Y, Liu T, Meyer CA, Eeckhoute J, Johnson DS, et al. (2008) Model-based analysis of ChIP-Seq (MACS). Genome Biol 9: R137.
5. Bailey TL, Elkan C (1994) Fitting a mixture model by expectation maximization to discover motifs in biopolymers. Proc Int Conf Intell Syst Mol Biol 2: 28-36.
6. Heinz S, Benner C, Spann N, Bertolino E, Lin YC, et al. (2010) Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and B cell identities. Mol Cell 38: 576-589.
7. Giannopoulo E, Elemento O (2011) Characterizing ChIP-seq peaks using customizable analysis workflows (submitted).
8. Li H, Durbin R (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics 25: 1754-1760.
9. Elemento O, Slonim N, Tavazoie S (2007) A universal framework for regulatory element discovery across all genomes and data types. Mol Cell 28: 337-350.
10. Quinlan AR, Hall IM (2010) BEDTools: a flexible suite of utilities for comparing genomic features. Bioinformatics 26: 841-842.
11. Mikkelsen TS, Xu Z, Zhang X, Wang L, Gimble JM, et al. (2010) Comparative epigenomic analysis of murine and human adipogenesis. Cell 143: 156-169.
12. O'Geen H, Lin YH, Xu X, Echipare L, Komashko VM, et al. (2010) Genome-wide binding of the orphan nuclear receptor TR4 suggests its general role in fundamental biological processes. BMC Genomics 11: 689.
13. Nielsen R, Pedersen TA, Hagenbeek D, Moulos P, Siersbaek R, et al. (2008) Genome-wide profiling of PPARgamma:RXR and RNA polymerase II occupancy reveals temporal activation of distinct metabolic pathways and changes in RXR dimer composition during adipogenesis. Genes Dev 22: 2953-2967.
14. Lefterova MI, Steger DJ, Zhuo D, Qatanani M, Mullican SE, et al. (2010) Cell-specific determinants of peroxisome proliferator-activated receptor gamma function in adipocytes and macrophages. Mol Cell Biol 30: 2078-2089.
15. github SOCIAL CODING: [http://github.com/pezmaster31/bamtools]

GRID distribution supports clustering validation of large mixed microarray data sets



Angelica Tulipano[1], Carmela Marangi[4], Leonardo Angelini[3], Giacinto Donvito[2], Guido Cuscela[2], Giorgio Pietro Maggi[2,3], Andreas Gisel[1§]

[1]CNR, Istituto Tecnologie Biomediche Sezione di Bari, Bari, Italy
[2]INFN Sezione di Bari, Bari (Italy),
[3]Dipartimento Interateneo di Fisica, Università degli Studi e Politecnico di Bari, Bari, Italy
[4]CNR, Istituto per le Applicazioni del Calcolo Sezione di Bari, Bari, Italy
[§]Corresponding author

Depicted authors have names underlined.

Abstract

Microarray data are a rich source of information, containing the collected expression values of thousands of genes for well-defined states of a cell or tissue. Vast amounts of data (thousands of arrays) are publicly available and ready for analysis, for example to scrutinise correlations between genes at the level of gene expression. The large variety of arrays available makes it possible to combine different independent experiments to extract new knowledge. Starting with a large set of data, relevant information can be isolated for further analysis. To extract the required information from data-sets of such size and complexity requires an appropriate and powerful analysis method. In this study, we chose to use an unsupervised hierarchical clustering algorithm, Chaotic Map Clustering (CMC), in a coupled two-way approach to analyse such data. However, the clustering approach is intrinsically difficult, both in terms of the unknown structure of the data and interpretation of the clustering results. It is therefore critical to evaluate the quality of any unsupervised procedure for such a complex set of data and to validate the results, separating those clusters that are due simply to noise or statistical fluctuations. We used a resampling method to perform this validation. The resampling procedure applies the clustering algorithm to a large number of random subsamples of the original data-matrix and, consequently, the whole process becomes computationally intensive and time consuming. Using Grid technology, we show that we can drastically speed up this process by distributing the clustering of each matrix to a separate worker node, and thus retrieve resampling results within a few hours instead of several days. Further, we offer an online service to cluster large microarray data sets and conduct the subsequent validation described in this paper.

Introduction

Today, biologists can measure the expression levels of thousands of genes under different experimental conditions using DNA microarray chips. A typical biological experiment produces different data-sets of expression values, monitoring the experimental conditions applied to a tissue or a cell culture. Often, after in-house analysis, researchers deposit their raw data with one of the available public repositories, from where they can be retrieved for different kinds of analysis. Public repositories, such as GEO [1] and ArrayExpress [2], have already collected vast quantities of various microarray experiment results. Such repositories allow independent data sets of expression values, obtained with the same chip design but in different conditions and in different laboratories, to be combined and compared. Studies on such enormous amounts of heterogeneous data can reveal new and significant information, and represent an interesting and important approach that can provide new insights into the behaviour of genes. Of course, this is a difficult task, because it requires an appropriate data-normalisation process and, even more important, an efficient method of analysis, such as clustering. Because of the heterogeneity of the data sets, supervised clustering methods and parametric algorithms are unsuitable, as nothing is known *a priori* about the structure and distribution of the data. The simplest way to analyse large and mixed data-sets without any loss of information is to use an unsupervised clustering method that requires neither *a priori* assumptions about the data nor a cut-off above a fixed threshold expression value. This gives us the possibility of discovering unknown correlations between genes or unexpected behaviours in different experimental situations. Chaotic Map Clustering (CMC) [3], which we have tested successfully for microarray data (unpublished data), was the unsupervised clustering algorithm chosen. This unsupervised method of analysis of a full non-restricted data set naturally produces a lot of noise, necessitating a very accurate procedure for validation of the clustering results. We have to be able to evaluate the quality of the results and to determine the optimal settings for the clustering procedure. Resampling, based on a cross-validation method [4], is an efficient way to evaluate clustering results, telling us if they are really due to a strong correlation between

genes or if they arise simply as a result of statistical fluctuations or noise. The resampling method requires the creation and clustering analysis of random subsets of the original data-matrix. To obtain statistically relevant results requires the creation and analysis of tens of resampled matrices. With matrices such as our test-set, with a size of 22,215 x 587, the resampling validation procedure with our clustering algorithm becomes a very memory-intensive and time-consuming computational process. It is crucial to speed up this procedure in order to perform this challenging microarray data analysis within a reasonable time frame. Grid technology gives us the possibility to split and distribute the resampling procedure over different processors, allowing us to evaluate the quality of the clustering research in a few hours rather than several days. In this way, we can efficiently analyse any mixed data set, and even increase the number of experiments included within the analysis process, thus allowing analysis of even larger matrices of microarray expression values.

Chaotic Map Clustering of microarray data

As a test-set for our clustering approach, we selected and downloaded from GEO a data-collection derived from the Affymetrix microarray design 'Human Genome U133 Array Set' (HG-U133A). This heterogeneous data-set includes 587 data samples from different laboratories and covers more than 20 different biological experiments relating to 22,215 different genes. We did not set any threshold for expression values, but considered the whole distribution of the data, from the lowest to the highest value, to be informative. Microarray data are intrinsically noisy owing to the procedure of measurement itself, but we maintain that even low expression values can have high information content, especially in such a mixed and large comparison between different biological experiments. Background and noise data are, in any case, evaluated in a second round (see below) by means of the whole procedure of clustering and validation of the results.

In order to have a comparable set of data, we scaled each data-set point by means of global normalisation, carrying out a logarithmic transformation and setting the median of the distribution of expression values of each microarray experiment to zero. We organised the data into an expression matrix, $D = N \times S$, where N (=22,215) is the total number of genes in the array design, and S (=587) is the number of samples (experiments).

Generally speaking, the clustering process aims to investigate and discover the unknown structure of a set of data by grouping objects that are more similar to each other according to some similarity measure. In this specific context, clustering microarray data can reveal groups of genes with similar gene-expression profiles that are co-regulated in different samples or groups of experiments.

To analyse data with no *a priori* knowledge of their structure (*i.e.*, the number of classes, or the geometric distribution), we chose an unsupervised hierarchical clustering algorithm: Chaotic Map Clustering (CMC) [3], which we have tested extensively on the clustering and analysis of heterogeneous microarray data sets (personal communication A. Tulipano).

Furthermore, to discover unknown relationships between genes within subsets of experiments that could be hidden by the signals of other genes, a coupled two-way approach [5] was applied using CMC. The first approach considers the samples as the objects to be clustered; the other considers the genes as the objects to be clustered. Using the groups of genes and samples obtained with two-way clustering, this method identifies submatrices of the total expression matrix on which further analysis can be performed locally with the user's preferred analytical tools, revealing new partitions of samples and genes and leading to new information. In this manner, by focusing on small subsets, we lowered the noise induced by the other samples and genes, and were able to discover partitions and correlations that were masked or hidden when the full data set was used in the analysis.

Details of the results, in terms of their biological relevance, are beyond the scope of the present paper and are thus not discussed further here. The focus of this report is the validation of the clustering results: we had to search for new solutions because the validation of such large data sets became computationally very intensive.

Cluster validation

In order to evaluate the whole procedure of clustering, selecting the optimal settings for the

algorithm, and to validate the vast number of clusters found by the process, we used a cluster-validation method based on resampling [4]. This is a cross-validation procedure, where subsets of the data-matrix under investigation, whose sizes are $fN \times S$, where $0 < f < 1$ is the reduction factor, are constructed randomly, and the clustering algorithm is applied to each subset. From these results, we created an $N \times N$ connectivity matrix T, for each resampled matrix, whose elements are:

$$\mathbf{T}_{ij} = \begin{cases} 1 & \text{points } i \text{ and } j \text{ belonging to the same cluster} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathrm{T}_{ii} = \begin{cases} 2 & \text{points } i \text{ is present in the resample} \\ 0 & \text{otherwise} \end{cases}$$

and compared it to the connectivity matrix of the original data-matrix.

Starting from the overlap of the original and resampled connectivity matrices, we can define three quantities [6], namely "sensitivity" (*sens*), "specificity" (*spec*) and "positive predictive value" (*ppv*), which can be regarded as useful "quality measures" of a clustering result.

To define these quantities, we considered the results obtained on the full size data-set as the "truth". According to the "truth", we have two classes: either gene i and gene j are in the same cluster (positive) or not (negative). We then compare the results obtained through resampling.

Table 1 lists all possible combinations: true positive (TP)—ij are in the same cluster, both in the original and in the resampled data-set; false negative (FN)—ij are in the same cluster in the original data-set, but not in the resampled set; true negative (TN)—ij are not in the same cluster, either in the original data-set or in the resampled one; false positive (FP)—ij are in the same cluster in the resampled matrix, but not in the original matrix.

It is now possible to define *ppv* as the average, with respect to the resamples, of the number of TP pairs divided by the number of the pairs belonging to the positive class

$$ppv = \left\langle \frac{N_{TP}}{N_{TP}+N_{FP}} \right\rangle (1)$$

and, with similar notation, *sens* is defined as

$$sens = \left\langle \frac{N_{TP}}{N_{TP}+N_{FN}} \right\rangle (2)$$

and *spec* is defined as

$$spec = \left\langle \frac{N_{TN}}{N_{TN}+N_{FP}} \right\rangle (3)$$

Evaluating these quality measures, we are able to identify stable clustering solutions, which are less likely to result from noise or statistical fluctuations, and can also evaluate the efficiency of the set of resolution parameters used for clustering.

To validate the clusters obtained by applying the CMC clustering algorithm to the original matrix of 22,215 x 587, 50 randomly resampled matrices of 16,661 x 587 (*i.e.*, a reduction factor of 25%) were generated. Each matrix was then clustered using CMC with the same set of resolution parameters. Clustering a single matrix of such a size with CMC is an intensive computational process that requires more than 1.5 GigaBytes of RAM, and takes about 2 hours of computing time (one CPU Xeon 3.0GHz). Moreover, the creation of the connectivity matrix and its comparison with the original matrix takes several hours. Clustering the whole set of resampled matrices of at least 50 random matrices and calculating the overlap of the connectivity matrices would occupy one single CPU for more than two weeks. With this method, cluster validation would be a slow and inefficient procedure. Splitting the whole cluster-evaluation process of the resampled matrices into several jobs, one for each resampled matrix, and distributing them on several CPUs would speed up the whole validation procedure, allowing quicker evaluation of the quality of our clustering.

Grid distribution

Because of the enormous quantity of computer resources available, the Italian Grid Infrastructure (Virtual Organisation bio) [7] provides the possibility of splitting a large, complex application into many smaller jobs that can run in parallel, greatly reducing the time needed to reach the final results. Our resampling process is easily divisible into many smaller processes, namely every randomised matrix of the resampling process can be launched as an independent job.

The Job Submission Tool (JST, [8]), developed to submit and monitor a large number of jobs (in the range of hundreds of thousands) in an almost automatic way, is the engine that runs and controls the workflow to cluster large microarray

data-sets and run the necessary validation. As a test of the efficiency and functionality of our workflow (see 'Workflow Implementation and Results' below), we used the above mentioned HG-U133A test-set of 587 microarray data from Affymetrix (22,215 genes), and analysed 50 randomised matrices, 16,661 x 587 in size, for the validation. The main concept of the JST is the "task". A certain number of tasks have to be executed to complete the challenge. The entire problem is first divided into steps that depend on each other, and then each step is, if possible, subdivided into the smallest possible independent elements that can be sent to the Grid. Depending on the time required for each element, a task can consist of one or several elements, in order to optimise the performance by balancing the time needed for installation and processing. Our implementation consists of two main steps and a final step, summarising the results, the first step being clustering of the main microarray data-matrix and the second, consecutive and dependent, step concerning cluster validation. Whereas the first and last steps are single events, the second can be subdivided easily into independent elements for further processing. CMC validation for such a large matrix takes several hours, and therefore each task contains one element – a matrix. The tasks are then routed to a central database (DB) server in order to assign each task to a Grid job; the server then takes full control of challenge completion. A robot is used to submit jobs to the Grid, to Worker Nodes (WN) that are initially identical and do not know which task they have to execute. Only when the job arrives at, and starts to be executed on, a WN does it request a task to be sent from the central DB. Every job asks the central DB for a task that has not yet been assigned to any other job ("free" task). Information on the execution of each task is logged in the central DB to maintain an overview of the processing. As soon as a job submitted by the User Interface (UI) to an available WN receives the task to be executed, it starts to download from a storage element all the files (input data and libraries) required to perform the task – in our case the clustering of a single matrix and running the validation.

Only when all steps are executed correctly is the status of a particular task on the central DB updated to "Done", and the results made available on a Web server for download. In this way, the central DB monitors task execution. No manual intervention is required to manage the re-submission of failed tasks. In fact, tasks that are found in a "running" state after a fixed time interval are considered to have failed and are automatically reassigned to new jobs. As soon as no task can be assigned to the submitted jobs, meaning all tasks are labelled "Done", the robot stops submitting jobs and the processing is terminated. In this way, the architecture of the JST allows highly effective and reliable exploitation of all the computational resources available on the Grid.

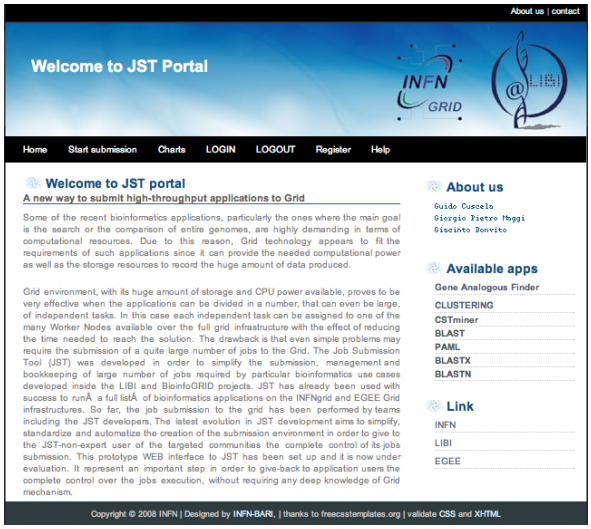


Figure 1. The JST portal.

Recent improvements in the JST include the implementation of a GUI (Graphic User Interface) in order to make this tool available to the bioinformatics community. The GUI (Figure 1) is available on a website [9], where users can register and, after authorisation, submit their applications to the Grid. Through the interface, the user defines parameters for the clustering application, and provides the input files that the JST will then elaborate to create the task list.

Workflow implementation and results

To orchestrate all the steps necessary to validate the initial clustering through the JST interface, we created a workflow to cluster the original matrix, as well as to create and cluster the randomised matrices. The results were then compared with the result of the original clustering to calculate the values of *ppv*, *sens* and *spec* (Figure 2).

After login and opening the submission window (see Figure 1, 'Start submission'), JST offers a list of 'gridified' applications; choose "CLUSTERING". The input schema (see Figure 3) requires 4 types of input information: (1) one input file as a text-file containing an expression matrix, where rows refer to genes, and columns to the normalised values of the microarray data-set; (2) the number of re-samplings, which establishes how many random matrices have to be produced and analysed by the CMC; (3) the distance measuring method (Euclidean or Pearson correlation); and (4) the indication whether to cluster by rows (genes) or by columns (experiments). This last parameter is important if a user wants to apply the coupled two-way approach and re-send the first clusters for the second clustering process. Before sending the submission to the Grid, JST displays all the parameters and the Grid submission commands for eventual control; only by confirming the submission at this stage does JST start to process the clustering and validation request.

The first step of the JST process consists of a single task, starting with the clustering of the original matrix and generation of the corresponding connectivity matrix. A perl script, distributed by the JST, provides the conversion of the input matrix file in a format that is accepted by the CMC clustering algorithm. Then it launches the clustering process and finally calculates the connectivity matrix of the clustering results. This step produces two outputs: a reference file of the clustering results and a text file of the connectivity matrix, both stored in a common repository. Because this connectivity matrix is needed for the tasks in the second step, JST has to make it available to every job responsible for executing these tasks. The best approach to achieve this goal is to register the file on a Grid Storage Element and to store the location in the central JST database so that every job can copy it locally in a secure and efficient way. After the first task is executed correctly, and the status of the related entry in the database is update to "Done", the Grid jobs can begin executing the second step and distributing the tasks of the second step (*i.e.*, validation). The second step consists of the random generation of each resample D_k, $k=1...m$ of the original data, the clustering by the CMC algorithm of this randomised matrix, generation of the related connectivity matrix T_k, comparison with the connectivity matrix of the original matrix,

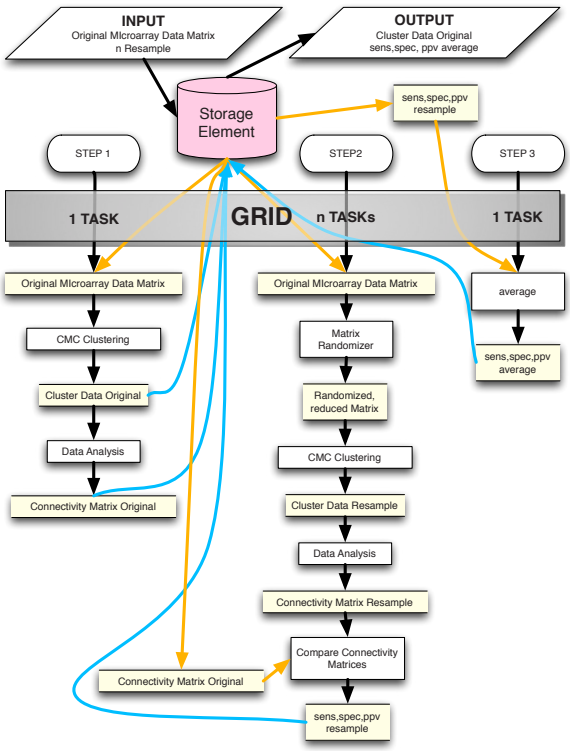


Figure 2. Distribution of processing over Grid nodes.
The first step of the JST process consists of a single task, starting with the clustering of the original matrix and generation of the corresponding connectivity matrix that is stored in the storage element. The second step consists of the random generation of each resample of the original data, clustering of this randomised matrix, generation of the related connectivity matrix, and computation of the *sens*, *spec* and *ppv* values. The final step retrieves all the resampling results from the storage element and calculates the average values of *ppv*, *sens* and *spec* for the whole set of resampled matrices. After all tasks are completed, the user can retrieve (from a Web link) two outputs: the clustering results of the original expression matrix, and the averaged values of the *ppv*, *sens* and *spec* file.

and computation of the quantities listed in Table 1. Again, a perl script, developed from our side and distributed by the JST, processes all these steps in order to realise the second part of the workflow. The system stores, in a common repository, one output file for each task, corresponding to a resampled matrix, with the related values of *ppv*, *sens* and *spec*. The final step retrieves all the resampling results and calculates the average values of *ppv*, *sens* and *spec* for the whole set of resampled matrices by means of a third perl script specifically implemented for this task. After all steps are completed, the user can retrieve (from a Web link sent by e-mail) two outputs: the clustering results file of the original expression matrix (step 1) and the averaged values of *ppv*, *sens* and *spec* file (step 3).

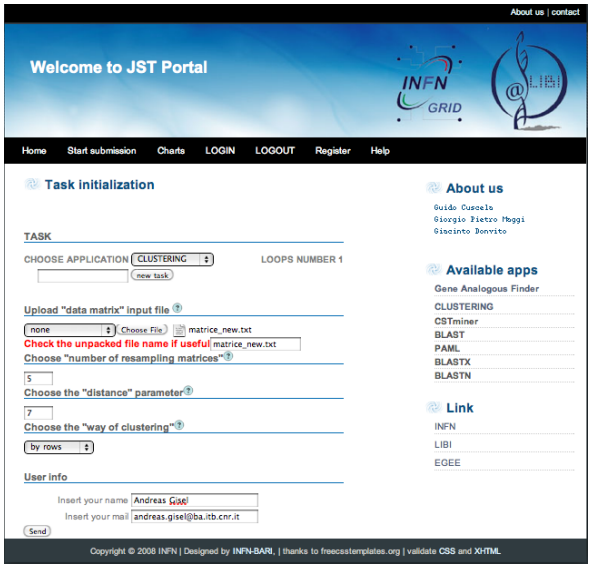


Figure 3. Input parameters.
The user uploads one input file containing genes (in rows) and the experiments of the microarray data-set (in columns), and sets the number of resamplings to execute and the method of distance measure (Euclidean or Pearson correlation).

For the test-case used here, a data-matrix of dimension of 22,215 x 587 and validation using 50 randomised, reduced matrices, distributed over the Grid (Figure 1) using one WN for each matrix to be analysed, we were able to reduce the processing time by 20-fold, from 16 days on a single CPU to 20 hours distributed over the Grid. The average execution time was about 8 hours per job. Considering that we have two steps, with the second being dependent on the result of the first, we have a net processing time of 16 hours, losing therefore about 4 hours of the total process time in job queuing. Nine matrices had to be resubmitted because of various different problems. One problem was the memory requirement (>1.5 GB RAM) of the clustering algorithm for a matrix of the specified size, which was a requirement that not all available WNs could satisfy.

The average values of *ppv*, *sens* and *spec* calculated for the given data set were 0.65, 0.81 and 0.95, respectively. Owing to the size of the data set, and the high level of fragmentation (more than 300 clusters), we can expect that the number of true negatives is orders of magnitude greater than the other quantities defined in Table 1. This means that, for the case at hand, the specificity values would be close to 1 even in the case of random or incorrect clustering results. For our purposes, we thus consider only *ppv* and *sens* as the relevant quality measures.

To illustrate the results in more detail, Figures 4 and 5 show the distributions obtained for the values of *ppv* and *sens* for the 50 resampled matrices. We can see from the histograms that about half the resamples exhibit sensitivity values above 0.95, and that, in almost all cases, *ppv* values are above 0.5. As it is well known that the statistical significance of the quality measures is affected strongly by the size of the data set, and by the level and nature of noise in the data, it has long been recognised that there is substantial intrinsic noise contained in microarray data. We stress that the values obtained here far exceed what is generally considered a good result in such a context, and we thus conclude that our results validate the approach proposed here.
Incidentally, we recall that re-sampling procedures can also be applied to search for an optimal set of algorithm parameters. In this case, resampling procedures need to be repeated for any choice of the clustering parameters, and records of the corresponding quality measures have to be collected. The optimal values of the parameters are then selected as those corresponding to the maximum value of *spec* and/or *ppv*. However, as we have already tested (personal communication A. Tulipano) the robustness of the algorithm against changes in the main scale parameter α, we limited ourselves to performing the resampling in order to assess the quality of the clustering results. However, we expect that keeping the scale parameter fixed on both the original and resampled data-set will add a slight negative bias to the *ppv* measures by increasing the number of false positives.

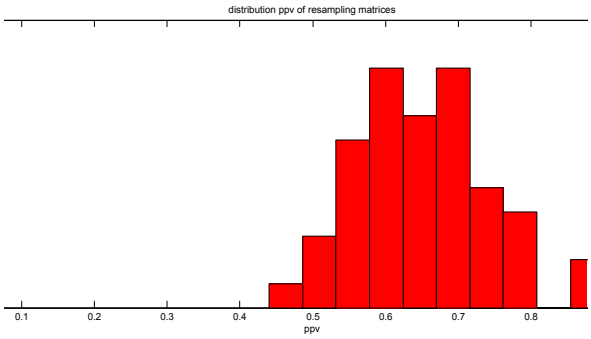


Figure 4. Distribution of ppv values obtained for the resampled matrices.

Conclusion

The approach described above, that is to distribute the process of validating clustering results on the Grid, proved to be a very valid implementation for large microarray data sets. The analysis of large mixed microarray data with the CMC algorithm is a very complex procedure, requiring an efficient method of result validation. The two key functionalities of our approach are the use of the JST, modified for this application, which has already been used by other applications to distribute workload efficiently over the Grid [8], and the newly developed three-step workflow to calculate and process the resampled matrices in parallel. Because the matrices are so large, and the clustering algorithm and associated procedure of evaluation is computationally intensive, the original matrix and every resampled matrix was submitted to an independent WN, allowing the total validation to be run in parallel, completing the calculation in a fraction of the original time required when using only one CPU.

This drastic improvement in the validation process will allow researchers to analyse any combination of available data sets with almost no limit to the data size and number of resampling cycles, to guarantee reasonable accuracy of the analysis and validation. In addition, the pipeline allows users not only to run time-consuming validation processes, but also provides users with the clustering data, and can therefore be used to run time-consuming clustering on large data-sets using CMC.

The only problem we were confronted with was the large amount of RAM required by the CMC clustering algorithm operating on a large dataset. However, very few WNs were not adequately equipped to execute the clustering algorithm. In such cases, the JST resubmitted those failing jobs to new WNs without requiring any user interaction, providing the end user with a complete and accurate picture for further analysis.

Authors' contributions

AT was responsible for the CMC algorithm, its implementation, the development of the validation process and preparation of the manuscript. CM was involved in the implementation of the CMC algorithm, implementation of the validation process and preparation of the manuscript. LA was responsible for the development and implementation of CMC algorithm. GD was responsible for the implementation of the Grid distribution and development of the graphical interface. GC was responsible for the implementation of the Grid distribution, development of the graphical interface and preparation of the manuscript. GPM was responsible for the implementation of Grid distribution. AG was responsible for the whole project, implementation of the workflow and preparation of the manuscript.

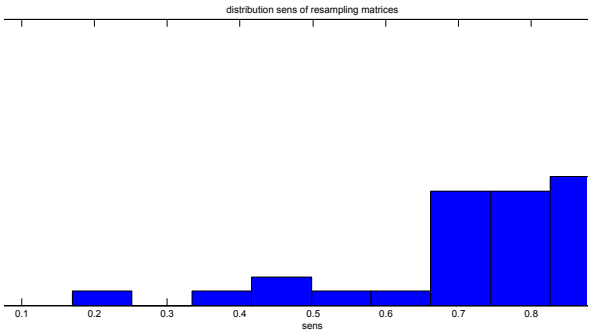


Figure 5. Distribution of sens values obtained for the resampled matrices.

Acknowledgements

The authors thank Prof. Mario Pellicoro for fruitful discussions about the implementation of the CMC algorithm and Nicola Losito for technical support. We also thank Dr. Helen Rothnie for the English corrections. We would also like to thank the INFN Grid production Resource Centre's administrators and the INFN-GRID/GRID.IT management. This work was funded for INFN Bari by the MIUR (Italian Ministry for Education, University and Research) in the LIBI "International Laboratory of BioInformatics" project, F.I.R.B. 2003, under grant RBLA039M7M; and for ITB by the EU project EMBRACE, which was funded by the European Commission within its FP6 Programme, under the thematic area 'Life sciences, genomics and biotechnology for health', contract No. LUNG-CT-2004-512092.

Competing interest statement

None declared

References

1. http://ncbi.nlm.nih.gov/geo.
2. http://www.ebi.ac.uk/arrayexpress.
3. Angelini L, De Carlo F, Marangi C, Pellicoro M, Stramaglia S (2000) Clustering Data by Inhomogeneous Chaotic Map Lattices. Phys Rev Lett 85: 554-557.

4. Levine E, Domany E (2001), Resampling method for unsupervised estimation of cluster validity. Neural Comp 13: 2573-2593.
5. Getz G, Levine E, Domany E (2000) Coupled two-way clustering analysis of gene microarray data. Proc Natl Acad Sci U S A 97: 12079-12084.
6. Van deer Laan MJ, Bryan J (2001) Gene expression analysis with the parametric bootstrap. Biostatistics 2: 445-461.
7. http://www.italiangrid.org/
8. De Sario G, Tulipano A, Donvito G, Maggi G, Gisel A (2009) High-throughput Grid computing for Life Sciences. In: M. Cannataro editor. Handbook of Research on Computational Grid Technologies for Life Sciences, Biomedicine, and Healthcare.
9. http://webcms.ba.infn.it/~jst/JST/

The future of HOPE: what can and cannot be predicted about the molecular effects of a disease causing point mutation in a protein?

Francesca Camilli[1], Annika Borrmann[1], Shima Gholizadeh[1], Tim te Beek[2], Remko Kuipers[3], Hanka Venselaar[1]

[1]CMBI, NCMLS, Radboud University Nijmegen Medical Centre, Nijmegen, Netherlands,
[2]NBIC, Netherlands Bioinformatics Centre, Nijmegen, Netherlands,
[3]Laboratory of Systems and Synthetic Biology, Wageningen University, Wageningen, Netherlands

Depicted authors have names underlined.

Abstract

Next generation sequencing is greatly speeding up the discovery of point mutations that are causally related to disease states. Knowledge of the effects of these point mutations on the structure and function of the affected proteins is crucial for the design of follow-up experiments and diagnostic kits, and ultimately for the implementation of a cure. HOPE can automatically predict the molecular effects of point mutations. HOPE does this by massively collecting highly heterogeneous data related to the protein and the mutated residue followed by automatic reasoning that as much as possible mimics the thinking of a trained bioinformatician. We discuss HOPE and review today's possibilities and challenges in this field.

Availability: HOPE is running as a web server available at www.cmbi.ru.nl/hope/

Introduction

The development of next generation sequencing (NGS) technologies is accompanied by a series of challenges ranging from problems with storage of large amounts of data to the under-

standing of all pathways and mechanisms in an organism [1]. One of these new challenges is the analysis and prioritisation of putative disease-causing point-mutations in human genetics studies. It has been estimated that single nucleotide polymorphisms (SNPs) occur as frequently as every 100-300 bases. This implies that in an entire human genome we can potentially find 10 to 30 million SNPs [2]. The publicly available Single Nucleotide Polymorphism Database (dbSNP) nowadays contains over 30 million variations of which over 12 million are located in genes [3]. A variant is called a SNP when it occurs in at least 1% of the population. This implies that most SNPs are not directly related to a serious disease because if they were, we would all be sick. Human genomes, however, also contain many rare variants and occasionally such a rare variant causes a serious disease.

NGS is revolutionising the way human geneticists search for the causative genetic defects for disease states. In the past, extensive family tree analysis (linkage analysis) would be followed by cloning and sequencing a small region of the human genome and subsequent bioinformatics studies of the genetic variants found in this region. Nowadays, human geneticists routinely sequence the entire exome and occasionally, even the entire genome of a patient. While sequencing a human genome will become cheaper, faster, and easier in the coming years, it will remain difficult to identify which of the many observed mutations are responsible for the phenotype/disease of interest. The rate at which genomes can currently be sequenced demands for an automatic approach towards the analysis and classification of newly found variants. When hundreds of variants are detected, they must be sorted in order of likeliness that they are causative for the disease studied; this process is commonly known as prioritisation.

Prioritisation consists of two steps when variants in the exome are being analysed. First the chance must be determined that the protein is related to the phenotype studied, and second, the chance must be determined that the mutation alters the function of that protein. The protein for which the product of these two chances is highest is the best candidate for follow-up studies. The first step, determining how likely it is that a protein is related to a phenotype, is the realm of system biology. The second step, determining how likely it is that a variant alters the function of a protein, is the ultimate goal of the HOPE software, the topic of this study.

We want to know if a variation in the patient's genome is harmless or possibly disease-causing. To do this we need to compare the variation found in our patient with the 'normal' human genome. As there is no such thing as the average human, we will have to compare the variations in our patient with the variations found in a large cohort of human genomes. These variations can be found in databases, such as dbSNP [3], and are described in the OMIM database [4]. By using information extracted from such databases we can classify the variations in our patient as either 'known to be harmless', 'known to cause a disease', 'previously found mutation with unknown effect' or even as a completely new mutation that is not present in the database(s) yet. A variations that is known to be harmless (often the ones that occur frequently in a population) can be removed from our list of putative disease-causing mutations. In case the variant matches an earlier described disease-causing mutation, there is no need for further investigation because the effect of the mutation is known already. Variants that fall in the categories 'unknown' and 'completely new' are worth further investigation.

The next step is to find out whether a mutation is located in the coding sequence of a gene, or in a regulatory sequence, splice site, or otherwise functional DNA. A mutation located in a regulation site might disturb the transcription or regulation of the gene, resulting in aberrant production of the protein. In contrast, a mutation located in the protein's coding sequence is likely to affect the folding of the protein instead of its production.

To really investigate the effect of variants on the protein we need to look at its 3D-structure. Studies of the mutation in 3D, can provide insight in the effect of the mutation and lead to ideas for experiments that eventually can result in a cure for the disease studied. Unfortunately, the Protein Data Bank (PDB) provides full or partial structures for only about 20% of all human proteins, while structural information for another 20% of the human proteins can be obtained using homology modelling techniques. This leaves a 60% of the sequences without known protein 3D-structure. To find more information about these proteins we need to rely on other information sources,

such as annotations and other information in databases, conservation scores from multiple sequence alignments, and predictions based on just the sequences. It is a time-consuming task to manually collect information from all these sources, combine them, and produce a coherent idea about the effect of the studied mutation. Not every (bio)medical researcher has the tools and the experience to work with bioinformatics databases, servers, and programs. More important is the fact that it is simply impossible to manually analyse every variant in the list that results from the NGS-run. An automatic approach is required.

A series of Web servers exist that can aid with the analysis of the effects of point mutations on a proteins structure and function. Table 1 lists many of these servers together with their present internet locations.

We believe that the analysis of disease related mutations should first of all include all smart ideas in the software listed in Table 1 but, additionally, should be open and extendible so that new ideas, new concepts, new data, etc., can be incorporated quickly. We also believe that the output of a mutation analysis server should be readable by life scientists, and not only by trained bioinformaticians. We therefore developed HOPE; a fully automatic program that can collect and combine all information available for a protein (including building a homology model when required) and produces a life scientist understandable report of the mutation at hand [15].

HOPE collects information from a wide range of information sources including calculations on the 3D-coordinates of the protein using WHAT IF Web services [16,17], sequence annotations from the UniProt database [18], conservation scores from HSSP [19], and predictions by a series of Distributed Annotation System (DAS) services [20]. When possible, homology models are built with YASARA [21]. Data is stored in a database and combined in a decision scheme to identify the effects of a mutation on the protein's 3D structure and its function. The decision scheme ensures that the most reliable source of information is used for the report, being first the 3D-structure, followed by the annotations in UniProt, that in turn

Table 1. Internet based web servers that can aid with the prediction of the effects of point mutations on a proteins structure and/or function. Left hand column: name of facility, reference, and URL. Right hand column: very short description of main feature. MSA (Multiple Sequence Alignment), SVM (Support Vector Machine), PSIC (Position Specific Independent Counts), GO (Gene Ontology).

Server + URL	**Main Feature**
SIFT [5] sift.jcvi.org/	Gives one score for tolerated or not, without explanation, based on MSA.
PolyPhen [6] genetics.bwh.harvard.edu/pph/	Gives damaging or not, uses annotations, info in PDB file, MSA, and PSIC [7] scores.
PolyPhen 2 [8] genetics.bwh.harvard.edu/pph2/	As PolyPhen but with simpler and better explained output. More visualisation options.
SNPs3D [9] www.snps3d.org/	Pre-calculated 3D-effects on known protein structures; visualization using Chime.
SNAP [10] rostlab.org/services/snap/	Gives neutral or non-neutral. Uses sequence annotations, predictions on sequence, and MSA.
Panther [11] www.pantherdb.org/tools/csnpScoreForm.jsp	Gives a score for deleterious or neutral based on MSA.
PhD-SNP [12] gpcr.biocomp.unibo.it/cgi/predictors/PhD-SNP/PhD-SNP.cgi	Predicts neutral/disease based on MSA using SVM.
PMut [13] mmb2.pcb.ub.es:8080/PMut/	Uses sequence based information and predictions but not structures. A set of pre-calculated mutations on a reference PDB set is also available.
SNPS&GO [14] snps-and-go.biocomp.unibo.it/snps-and-go/	Mainly uses GO terms to indicate disease versus neutral.

are followed by sequence-based predictions. The user can submit his/her sequence and mutation of interest via the web-interface. The report will be shown at the same website and is illustrated with figures and animations showing the effects of the mutation.

While HOPE has been shown to often provide very accurate descriptions of the expected effects of mutations, it most certainly also makes the occasional error and it has limitations in terms of which bioinformatics aspects of mutant analyses it can address. We validated the software by repeating a large number of mutation analyses we performed manually in recent years and by analysing mutations described recently in articles published in high quality journals. In these articles the authors describe their analysis of the structure of the protein and the structural and/or functional effects of the mutation(s). This extensive study revealed a few HOPE-improvements that we have already implemented, and potential improvements that for a series of reasons cannot be implemented yet. Surprisingly, it also revealed a number of instances in which the authors of peer-reviewed articles in highly respected journals made errors in the bioinformatics underlying their conclusions regarding the molecular effects of the mutation studied. We believe that HOPE can help the human genetics community by providing a 'second opinion' to referees, and perhaps also to the human geneticists publishing their results. However, it should be kept in mind that HOPE is software and thus equally fallible as a human being.

Method

In recent years we collaborated in numerous human genetics projects, performing the mutation analyses, providing insight in the structural effects of mutations and, in some cases, we could also provide suggestions for new experiments. Often these studies required building a homology model, but occasionally structure information could not be obtained so that the use of sequence-based prediction servers was required to obtain all information available for the protein. Here we use these examples to validate HOPE. To extend the validation beyond our own projects that, after-all, guided the design of HOPE, we decided to test HOPE using projects that were recently described in well-known journals, such as the American Journal of Human Genetics, Nature Genetics, and Human Mutation. These journals are known to contain many articles about disease-causing mutations and their structural effects. We selected a list of mutations and performed the analyses both manually, using YASARA for model building and visualization, and automatically, using HOPE for modelling and analyses. By comparing the results we obtain an overview of the strong and weak points of HOPE, and of features that can be improved or added to the system.

Results and discussion

The full results of the analyses can be found at the HOPE website[1] and are summarized in Table 2. We classified HOPE's result as 'good' when the HOPE report contained a clear and correct description of the effect of the mutation on the 3D-structure and/or function of the protein. A result received the classification 'OK' when it contained most but not all crucial points about the mutations and no erroneous remarks were found in the report. Of course, we want the report to be as complete as possible but it will take years before we can include every possible information-source. Therefore, possible points for improvement are mentioned in the last column of the Table. Results that were fully correct and did no teach us anything about possible HOPE improvements are not listed in the Table but are available at the website. Since March 2010 users from all over the world have visited the website more than 1600 times.

In these in-house studies we compared the manual and automatic analyses of 79 mutations in 26 proteins. The number of mutations per

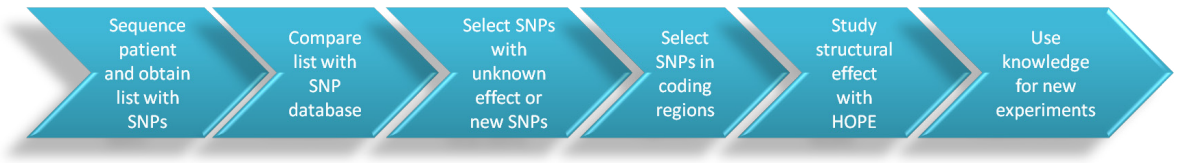


Fig 1. This figure shows how HOPE fits in the pipeline for SNP analysis.

1 www.cmbi.ru.nl/~hvensela/HOPEresults/

Table 2. Mutation analyses on in-house projects. Mutations shown in bold were explained using an experimentally solved structure, mutations in grey indicate the ones for which neither a structure nor a modelling template was available. The remaining mutations were explained using a homology model.

Protein (UniProt accession code) and reference	Mutation	HOPE's performance/points for improvement
HFE _ human PMID:18042412	**H63D** **G93R** **I105T** **L183P** **C282Y**	Good, using the complex structure and indirect interactions with other molecules could improve the results
EHMT1 _ Human PMID:19264732	**C1042Y** **R1166W**	Good, indirect dimer-interactions and a 'does the rotamer fit'-option could improve the results[1]
NDP _ Human PMID:20340138	C55R G67E G67R F89L S92P P98L K104N	As good as possible without a model, C55 is predicted buried which is not underlined by a low-res model, missing literature info for F89, the low resolution model also has info about putative cysteine bonds in the vicinity of S92 and P98
TOMT _ Human PMID:18953341	R81Q W105R E110K	Good, could benefit from a ligand contact analysis that also includes neighbouring residues
PO3F4 _ Human PMID:19671658	R329P	Good, analysis of contacts made with neighbouring residues could improve the answer
TMPS6 _ HUMAN	C702F R774C	Good, misses a possible new cysteine-bond for R774C
NDUF3 _ Human PMID:19463981	M1T G77R R122P	Good, but almost no info for M1T, result for R122P could benefit from knowledge about active site locations[1]
SEC63 _ Human PMID:20095989	I120T / D168H R217C / R267S Q375P / W651G D675E	HOPE's model differs from the manually built model and results in different accessibility-scores for D168. Others are OK
GLU2B _ Human PMID:20095989	R139H K155R M175V T261S R281W E381K	As good as possible without a model. A helical wheel predictor to identify the hydrophobic side of the helix could improve the result for E381K
KCNA1 _ Human PMID:19903818	N255D	OK, could benefit from literature information about the location of the voltage sensor.
TRPM6 _ Human PMID:18490453	G1955A	OK, could benefit from information about the glycine-rich motif.
NDUV1 _ Human	L53P/ P122L Y204C / C206G A211V / R257Q A341V / T423M	Good, results for P122L could be improved using the complete complex for analysis and information about residue stacking for R257.
NDUS2 _ Human	F84L E104G R228Q P229Q S413P D446N	OK, slight difference between the manual and automatic model, could benefit from using the complex for R228Q and information about the location of the membrane.
NDUS8 _ Human	P79L R94C R102H	Good, R94 could benefit from using the complex structure.

1 The 'does the mutated residue fit' option has been implemented as a result of this validation experiment. Additional WHAT IF options that have been added to HOPE are 'does the mutated residue make hydrogen bonds', 'does the mutated residue make a salt bridge', and 'does the mutated residue influence the shape of a cavity'.

Table 2 (cont.)		
PCD15 _ Human PMID:18719945	R134G D178G G262D	Good, result for R134G could improve by using a model that covers more domains.
SMAD3 _ Human PMID:21217753	**T261I** **R287W**	Good, result for T261I could improve using the complex structure
TLR2 _ Human	T411I R579H P631H R753Q	OK, could be improved using long distance relations and neighbour analysis in the complex.
ACAD9 _ Human PMID:20816094	E413K R518H	Good, result for R518 could benefit from long distance relations.

protein ranged from one to eight, and the protein length ranged from a small single-domain protein of 133 residues to a large multi-domain protein of 2022 residues. Nine mutations could be explained using the experimentally solved structure of the protein or protein-domain while 53 mutations could be explained using a homology model. In some cases we had to build multiple models for a single protein because these domains were only available as separate templates. For 17 mutation studies no solved structure or template could be identified and therefore our analyses had to rely on sequence based predictions and annotations. HOPE uses a very safe homology modelling threshold to make sure that the models are build only when a good template is available. Consequently, HOPE does not identify a template for NDP _ Human. However, the cysteine pattern in the sequence indicates that the protein adopts a cysteine-knot fold [22]. We were able to manually build and use a homology model for NDP. This is the only project in which we used different information sources for the manual and automatic approaches.

To extend HOPE's validation beyond our in-house projects we also analysed mutations that were reported in the literature. The selection criteria used to select test-cases from the literature included: one or more mutations were found to cause a disease, a description of the structural effects of the mutation(s) given in the article, and, if possible, a description of the model building process. The results are summarised in Table 3. Again, we classified the results as good when HOPE was able to give a clear report that agrees with our manual analysis and that is as complete as possible, while 'OK' was used for correct but incomplete cases. As in Table 1, results are not listed here if they would not teach us anything about potential HOPE improvements.

We analysed 66 mutations in 32 proteins of which 27 could be explained using the experimentally solved protein structure, 32 could be explained using a homology model. For the remaining 7 mutations we used other information sources.

Sometimes, the protein of interest was solved multiple times under different conditions. This means that HOPE had to choose which of these PDB-files to use for the analyses and/or model building. A decision schedule in HOPE will decide which template/structure to use based on the length of the aligned sequence, the percentage identity, the resolution of the solved structure, and of course on the necessity that the mutated residue must be part of the model. It sometimes happens that the authors of the article decided to use a different PDB-file. In case of the KFL1-project, for instance, this makes sense because a better template for modelling was not solved until after the article was published. In other cases the authors used experimental knowledge that only they could have to decide on a certain PDB-file as template because it contains the protein in a certain state such as active/inactive, open/closed, or bound to a certain ligand. As a result, some of HOPE's analyses were performed using a different PDB-file but in most cases this did not affect the outcome of the analyses. The choice of PDB file or modelling template indeed is a point of concern; Bywater recently worded this problem nicely [23]. Should we model the inactive state or the active state? Actually, we should probably model both states because if a mutation influences either one of the two, it will already influence the protein's function.

Table 3. Mutation analysis on previously reported mutations. Columns and fonts as for Table 2.

Protein + reference	Mutation	HOPE's performance
CHSTE _ Human PMID: 20004762	**R135P** **L137G** **R231P** **Y293C**	Good, could be improved using motif information from literature.
DPM3 _ Human PMID: 19576565	**L85S**	OK, could be improved using dimer-structure and a coiled-coil predictor.
CSKP _ Human PMID: 19200522	**R28L**	As good as possible, could be improved using splice-site analysis.
ALR _ Human PMID: 19409522	R194H	Good (almost no info in article at all)
ACTA _ Human PMID: 19409525	R39H R118C R149C R185Q R258C R258H	OK, could benefit from annotations about the location of the nucleic binding cleft, or a service that calculates this.
EMG1/NEP _ human PMID: 19463982	D86G	Good, could benefit from analysis of contacts made by neighbouring residues.
TRPV4 _ Human PMID: 19232556	D333G	As good as possible, ANK-repeat not annotated, protein becomes more active.
LRCC50 _ Human PMID: 19944405	L175R	Good, could be improved using information from literature.
RENI _ Human PMID: 21036942	**D38N** **S69Y**	Good, could be improved using a more extensive analysis of the contact residues (S69Y).
SPSY _ Human (SMS) PMID: 20556796	**G56S** **V132G** **I150T**	Good, improved the conclusions drawn by the authors.
KLF1 _ Human PMID: 21055716	E325K (better modelling template now)	OK, but HOPE misses possible new interactions formed after mutation.
FXRD1 _ human PMID: 20858599	R325W	Good, even though no model was built (template identity does not exceed HOPE's safe modelling threshold).
PSB8 _ Human PMID: 21129723	T75M	OK, could benefit from better annotation of the active site residues
PPA5 _ Human (ACP5/TRAP) PMID: 21217755	**T89I** **G215R** **D241N** **M264K**	Good, finds points not mentioned in the article
PRPS1 _ Human PMID: 20021999	**D65N** **A87T** **I290T** **G306R**	OK, could be improved using the complete hexameric biological unit (not in PDB).
ABCAD _ Human PMID: 19944402	T4031A	Good, could be improved using information about the motifs in literature.
DCTN1 _ Human PMID: 19136952	**G71A**	Good, could benefit from motif information found in literature.
PGDH _ Human PMID: 18500342	**A140P**	Could benefit dramatically from a better neighbour analysis.

In our test-cases HOPE did not make any dramatic mistakes. However, some of the, otherwise correct, answers can be improved by the implementation of new Web services, by a smarter

choice of modelling templates, or by the use of literature information. Table 4 shows a short summary of HOPE's strong points (green), points that will be improved in the (near) future (orange), and points that will not be improved soon (red). These points will be discussed more extensively below.

HOPE can: *collect structural information from the 3D-structure or build a homology model when required*

A protein's 3D-structure contains an enormous amount of useful information. The fact that HOPE builds and uses the homology model is one of its strong points because this doubles the percentage of human proteins for which 3D analysis is possible. The YASARA modelling script used in HOPE was one of the top-performers in the CASP 2008 and 2010 competition [24]. We have to keep in mind that every homology model represents only a prediction of the truth. However, the choice for one of the best modelling methods and the use of a safe homology-modelling threshold reduces the chance that HOPE analyses a completely wrong model.

HOPE can: *use the most reliable information source and combine them*

HOPE will always provide an answer. Even when there is hardly any information known about the protein, HOPE can still use predictions and information about the amino acids. The fact that HOPE uses structure, annotations, predictions, and conservation scores makes that HOPE gives more complete answers than most other servers that often use just one source of information.

HOPE can: *give a clear and understandable answer for everyone in the (bio)medical fields*

HOPE aims to serve a group of users in the field of life sciences that typically lack extensive bioinformatics experience. Therefore, the HOPE website and reports are as easy to use and understand as possible. Difficult bioinformatics keywords in the report are linked to our freely available online dictionary[2] that is based on WikiPedia's software[3].

HOPE will, in the near future, be able to: *choose the structure/templates for modelling*

Template selection is difficult but occasionally crucial. If the template includes an interaction partner, knowledge about disturbance of the interface can be gained. If an enzyme has an active and an inactive form, then both should be modelled and analysed as disturbing any of

Table 4. Summary of HOPE's strong and weak points.

HOPE can:	collect structural information from the 3D-structure or build a homology model when required
	use the most reliable information source and combine them (known structure/homology model, UniProt annotations, conservation scores, predictions)
	give a clear and understandable answer for everyone in the (bio)medical field
HOPE will, in the near future, be able to:	choose the structure/templates for modelling in a more intelligent way, keeping in mind that the protein might be solved in different conformations, complexed with different ligands, and/or under varying conditions
	use more information from new DAS servers or other sources
	analyse also the mutated situation and compare this to the wild-type, model or structure
	analyse long-distance relations
HOPE will not easily be able to:	use all information in the heads of the specialists all over the world
	to extract information from literature

2 www.cmbi.ru.nl/wiki/
3 http://www.mediawiki.org

the two states will disturb the function. Currently, HOPE can detect multimeric interactions in case the used PDB file contains a multimer. However, not every protein was solved in its biological assembly making it difficult to detect the interactions between the protein of interest and its partners, and transient interaction partners are only seldom co-crystallised. For example, for the mutations in SMAD3 [25] HOPE performs its analyses using the PDB file of the SMAD3 monomer (1mjs [26]) instead of the trimeric complex (1u7f [27]) because the first one has a significantly better resolution (1.91 vs 2.60 Å). Analyses of the mutation T281I could benefit from using the trimeric complex because this residue obviously makes protein-protein interactions at the trimer interface. In this example the interactions are not mentioned in the report, although HOPE does mention the possibility to form these interactions. In the future we want to improve the choice for templates/structures by incorporating biologically assemblies from the protein interaction database PISA [28] and by using smarter algorithms for structure choices.

HOPE will, in the near future, be able to: *use more information from new DAS servers or other sources*

Nowadays, we can access an ever increasing number of servers and databases that all provide useful information. The latest NAR special volume on databases [29] lists hundreds of databases that all might for one project or another contain useful information, but obviously today's technological possibilities preclude use of all these databases. We do intend to let the number of databases grow that HOPE can tap in to, but logistics and maintenance issues will limit us to dozens of databases rather than hundreds. HOPE could then use this information and would give a more detailed report including this domain information. The validation of HOPE gave us new ideas for possible structure calculations and prediction services that can be used to improve the reports. For instance, mutation R28L in the CSKP project [30] probably affects a splice site. As soon as there is a server that provides information about splice sites in protein sequences we can include this in the HOPE reports, and if nobody makes such a server, we will in due time (have to) do it ourselves. Mutation L85S in DPM3 [31] shows that a prediction of coiled-coil domains can be useful, especially since this seems to be the only information known about this position. We also found new ideas for WHAT IF calculations that would improve the results and a few of them have already been implemented in the system (like whether a residue is lining the wall of a cavity).

HOPE will, in the near future, be able to: *analyse also the mutated situation and compare this to the wild-type model or structure*

Insufficient detail is obtained when only the wild-type residue is analysed because the model of the mutant occasionally adds information. For instance, in the case of mutations in SPSY [32] we see that the mutant residue could possibly form new interactions thereby stabilising the protein structure and changing its behaviour. Currently, HOPE only finds the interactions that cause a loss of stability, not the ones that cause gain of stability. In the future we want to implement a module that looks for newly formed beneficial interactions. We already implemented, for example, a module that looks at the stabilising effects of prolines near the N-terminus of helices.

HOPE will, in the near future, be able to: *analyse long-distance relations*

All residues are in close contact with others. Mutation of one residue will thus also affect its spatial neighbours. For example, mutation R329P in PO3F4 [33] changes an arginine that forms hydrogenbonds an asparagine that binds directly to DNA. Mutation of this arginine can therefore indirectly affect DNA binding even though the residue itself was not found to contact DNA. In similar ways, mutations can affect ligand binding sites, active sites, etc., even when the residue itself is not found in such sites. To find these effects we plan to extend the HOPE modules with a neighbour-analysis module that considers all residues that make contact with the mutated residue. Analysis of distance relations that span more than two residues will remain difficult.

HOPE will not be able to: *use all information in the heads of the specialists all over the world*

Common knowledge obtained by years of experience in looking at protein structures cannot be stored in a database. In case of the R122P mutation in NDUF3 [34], experience tells us that the mutation is located at the same side where

you can usually find the active site in homologous proteins. Today there is no easy way, yet, to annotate this type of information.

HOPE will not be able to: *extract information from literature*

Unfortunately, there is lots of information in the literature that is not (yet) stored in an easy accessible database. Sometimes this can simply be solved by annotating the information in the UniProt database. For instance, the location of the voltage-sensor in KCNA1 or the G-motif in TRPM6 can easily be added to the sequence features of the UniProt-records for these proteins. Analysing the results of HOPE for almost a hundred published cases, we realised that a trained protein structure bioinformatician knows an amazingly large number of 'little facts'. Putting this all in software will require a few more years of programming artificial intelligence code. Machine learning will not be able to do this work for us, as the number of 'little facts' that need to be encoded is still very much larger than the number of well analysed disease causing human variants.

Example result

We would like to share one of our projects as an example of what can already be done by HOPE and what will need to be improved in the future. Two mutations were analysed in protein EHMT1; C1042Y and R1166W. The protein structure of the domain of interest was solved and can be found in PDB file 3hna [35].

By studying this protein structure we could see that the cysteine at position 1042 makes important interactions with one of the zinc-ions in the zinc-cluster in this protein. This cluster is probably important for stabilisation of the local structure

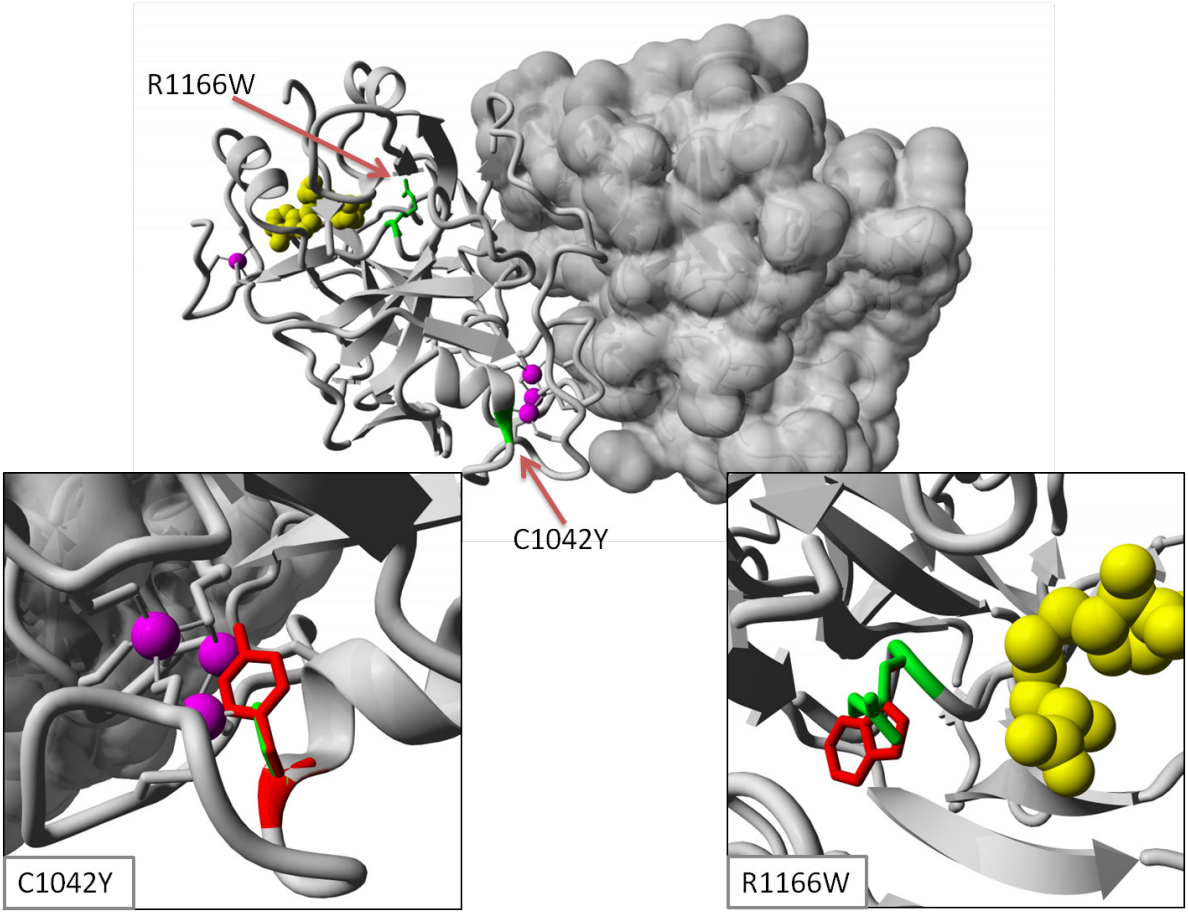


Figure 2. Overview of the mutations C1042Y and R1166W in the 3D-structure of EHMT1. The surface of only one of the monomers in the dimer is shown, the other monomer is depicted in cartoon representation. The mutated residues are colored green and indicated with red arrows. The Zinc-ions are shown as magenta balls while the ligand is shown in yellow balls. The insets show a close-up of the mutations. The side chain of the wild-type residue is now shown in green while the side chain of the mutant residue is shown in red.

element that seems needed to correctly position the loops that make interactions with the other monomer. The mutation will cause the introduction of a bigger residue which will simply not fit here and will therefore affect dimerisation. HOPE mentions the same points in its report: the interactions with the zinc-ion will be lost and the bigger residue will not fit at the same position. However, HOPE misses the fact that this could affect dimerisation because the mutated residue is not in direct contact with the other monomer. As soon as we have implemented a more extensive contact-analysis, HOPE will also be able to identify this effect.

The second mutation converts arginine 1166 into a tryptophan. In the protein structure we can see that this residue is buried and in contact with the ligand. We used a WHAT IF option to find out that no rotamer of tryptophan will ever fit at this position. The mutant will disturb the structure of the ligand binding site. HOPE also produces a report that mentions the interaction between R1166 and the ligand, that a bigger residue will probably not fit at the same position and that this will disturb interaction with the ligand. However, HOPE did not try to fit all possible rotamers of tryptophan. We are currently implementing this option.

This example shows that HOPE can already give a correct and informative answer that can be obtained easily and automatically. It also shows that there are still possibilities to improve the system and to provide even more clear and stronger answers.

We even found a few cases in which HOPE provided significantly more information than could be found in the article. For example, the authors of the ALR _ human project [36] performed a large number of experiments to find out that the mutation affects the function of the protein and might cause complex IV deficiency. HOPE mentions that the mutation is located in the ERV/ALR sulfhydryl oxidase domain and makes hydrogenbonds to FAD. The difference in size and charge will disturb this interaction which will in turn affect the function of the protein.

In this second example we show that HOPE can even improve results of mutation analyses. In their study of mutations in SPMSY causing Snyder-Robinson-Syndrome the authors describe mutation I150T. They used the experimentally solved structure of the SPMSY and found that the mutant residue threonine could make a new hydrogenbond with aspartate 222 in the hydrophobic core. We performed the same analysis but could not identify the same hydrogenbond. The minimum distance between the side chains of threonine and aspartate was found to be 4.5Å whereas a maximum distance of 3.5Å is required for hydrogenbond formation. It seems unlikely that this hydrogenbond is formed. HOPE produces a report that agrees with our manual analysis. This illustrates that HOPE can be used as engine to aid both authors and referees.

Conclusion

We have developed HOPE, a fully automatic mutant web server that can analyse the effect of point mutations on a protein's 3D-structure. We validated this server using a large number of well-described point mutations. We found that HOPE is able to give a clear and correct answer that in the majority of cases is similar to the results obtained by manual analysis. HOPE's performance depends on the information that is annotated or can be calculated from the structure. With this in mind, we think that HOPE performs very well in these projects providing clear and useful answers, even though they are not fully complete in some cases.

With the development of HOPE we have provided on small piece of the molecular puzzle: mutation analysis. It is now possible to automatically study the effects of point-mutations in the protein-coding region of the genome. In the future we can think of using HOPE to prioritise these mutations based on the probability that the mutation is disturbing the 3D structure and as such causing a disease. HOPE's analysis can then be added to the process of analysing the results from a NGS-run.

Competing interest statement

None declared

References

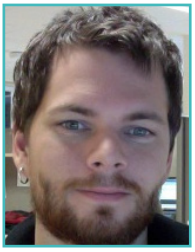
1. Gisel A, Bongcam-Rudloff E (2011) EMBRACE workshop "NEXT GENERATION SEQUENCING II". EMBnet.journal, 16(1):5-7.
2. The International HapMap (2003) Project. Nature, 426(6968):789-796.
3. Sherry ST, Ward MH, Kholodov M, Baker J, Phan L, Smigielski EM, Sirotkin K (2001) dbSNP: the NCBI database of genetic variation. Nucleic Acids Res 29(1):308-311.

4. Hamosh A, Scott AF, Amberger J, Valle D, McKusick VA (2000) Online Mendelian Inheritance in Man (OMIM). Hum Mutat 15(1):57-61.
5. Ng PC, Henikoff S (2003) SIFT: Predicting amino acid changes that affect protein function. Nucleic Acids Res, 31(13):3812-3814.
6. Ramensky V, Bork P, Sunyaev S (2002) Human non-synonymous SNPs: server and survey. Nucleic Acids Res 30(17):3894-3900.
7. Sunyaev SR, Eisenhaber F, Rodchenkov IV, Eisenhaber B, Tumanyan VG, Kuznetsov EN (1999) PSIC: profile extraction from sequence alignments with position-specific counts of independent observations. Protein Eng 12(5):387-394.
8. Adzhubei IA, Schmidt S, Peshkin L, Ramensky VE, Gerasimova A, Bork P, Kondrashov AS, Sunyaev SR (2010) A method and server for predicting damaging missense mutations. Nat Methods, 7(4):248-249.
9. Yue P, Melamud E, Moult J (2006) SNPs3D: candidate gene and SNP selection for association studies. BMC Bioinformatics 7:166.
10. Bromberg Y, Rost B (2007) SNAP: predict effect of non-synonymous polymorphisms on function. Nucleic Acids Res 2007, 35(11):3823-3835.
11. Thomas PD, Campbell MJ, Kejariwal A, Mi H, Karlak B, Daverman R, Diemer K, Muruganujan A, Narechania A (2003): PANTHER: a library of protein families and subfamilies indexed by function. Genome Res. 13(9):2129-2141.
12. Capriotti E, Calabrese R, Casadio R (2006) Predicting the insurgence of human genetic diseases associated to single point protein mutations with support vector machines and evolutionary information. Bioinformatics 22(22):2729-2734.
13. Ferrer-Costa C, Gelpi JL, Zamakola L, Parraga I, de la Cruz X, Orozco M (2005) PMUT: a web-based tool for the annotation of pathological mutations on proteins. Bioinformatics 21(14):3176-3178.
14. Calabrese R, Capriotti E, Fariselli P, Martelli PL, Casadio R (2009) Functional annotations improve the predictive score of human disease-related mutations in proteins. Hum Mutat 30(8):1237-1244.
15. Venselaar H, Te Beek TA, Kuipers RK, Hekkelman ML, Vriend G (2010) Protein structure analysis of mutations causing inheritable diseases. An e-Science approach with life scientist friendly interfaces. BMC Bioinformatics, 11:548.
16. Hekkelman ML, Te Beek TA, Pettifer SR, Thorne D, Attwood TK, Vriend G (2010) WIWS: a protein structure bioinformatics Web service collection. Nucleic Acids Res 38(Web Server issue):W719-723.
17. Vriend G (1990) WHAT IF: a molecular modeling and drug design program. J Mol Graph 8(1):52-56, 29.
18. Jain E, Bairoch A, Duvaud S, Phan I, Redaschi N, Suzek BE, Martin MJ, McGarvey P, Gasteiger E (2009) Infrastructure for the life sciences: design and implementation of the UniProt website. BMC Bioinformatics 10:136.
19. Schneider R, de Daruvar A, Sander C (1997) The HSSP database of protein structure-sequence alignments. Nucleic Acids Res 25(1):226-230.
20. Prlic A, Down TA, Kulesha E, Finn RD, Kahari A, Hubbard TJ (2007) Integrating sequence and structural biology with DAS. BMC Bioinformatics 8:333.
21. Krieger E, Koraimann G, Vriend G (2002) Increasing the precision of comparative models with YASARA NOVA--a self-parameterizing force field. Proteins 47(3):393-402.
22. Meitinger T, Meindl A, Bork P, Rost B, Sander C, Haasemann M, Murken J (1993) Molecular modelling of the Norrie disease protein predicts a cystine knot growth factor tertiary structure. Nat Genet 5(4):376-380.
23. Bywater R (2010) Solving the protein folding problems. Nature Precedings.
24. Krieger E, Joo K, Lee J, Raman S, Thompson J, Tyka M, Baker D, Karplus K (2009) Improving physical realism, stereochemistry, and side-chain accuracy in homology modeling: Four approaches that performed well in CASP8. Proteins 77 Suppl 9:114-122.
25. van de Laar IM, Oldenburg RA, Pals G, Roos-Hesselink JW, de Graaf BM, Verhagen JM, Hoedemaekers YM, Willemsen R, Severijnen LA, Venselaar H et al. (2011) Mutations in SMAD3 cause a syndromic form of aortic aneurysms and dissections with early-onset osteoarthritis. Nat Genet, 43(2):121-126.
26. Qin BY, Lam SS, Correia JJ, Lin K (2002) Smad3 allostery links TGF-beta receptor kinase acti-

vation to transcriptional control. Genes Dev 16(15):1950-1963.

27. Chacko BM, Qin BY, Tiwari A, Shi G, Lam S, Hayward LJ, De Caestecker M, Lin K (2004) Structural basis of heteromeric smad protein assembly in TGF-beta signaling. Mol Cell 15(5):813-823.
28. Krissinel E, Henrick K (2007) Inference of macromolecular assemblies from crystalline state. J Mol Biol 2007, 372(3):774-797.
29. Special database issue (2011) Nucleic Acids Res 39.
30. Piluso G, D'Amico F, Saccone V, Bismuto E, Rotundo IL, Di Domenico M, Aurino S, Schwartz CE, Neri G, Nigro V (2009) A missense mutation in CASK causes FG syndrome in an Italian family. Am J Hum Genet 84(2):162-177.
31. Lefeber DJ, Schonberger J, Morava E, Guillard M, Huyben KM, Verrijp K, Grafakou O, Evangeliou A, Preijers FW, Manta P et al. (2009) Deficiency of Dol-P-Man synthase subunit DPM3 bridges the congenital disorders of glycosylation with the dystroglycanopathies. Am J Hum Genet 85(1):76-86.
32. Zhang Z, Teng S, Wang L, Schwartz CE, Alexov E (2010) Computational analysis of missense mutations causing Snyder-Robinson syndrome. Hum Mutat, 31(9):1043-1049.
33. Lee HK, Song MH, Kang M, Lee JT, Kong KA, Choi SJ, Lee KY, Venselaar H, Vriend G, Lee WS et al. (2009) Clinical and molecular characterizations of novel POU3F4 mutations reveal that DFN3 is due to null function of POU3F4 protein. Physiol Genomics, 39(3):195-201.
34. Saada A, Vogel RO, Hoefs SJ, van den Brand MA, Wessels HJ, Willems PH, Venselaar H, Shaag A, Barghuti F, Reish O et al. (2009) Mutations in NDUFAF3 (C3ORF60), encoding an NDUFAF4 (C6ORF66)-interacting complex I assembly protein, cause fatal neonatal mitochondrial disease. Am J Hum Genet 84(6):718-727.
35. Wu H, Min J, Lunin VV, Antoshenko T, Dombrovski L, Zeng H, Allali-Hassani A, Campagna-Slater V, Vedadi M, Arrowsmith CH et al. (2010) Structural biology of human H3K9 methyltransferases. PLoS One, 5(1):e8570.
36. Di Fonzo A, Ronchi D, Lodi T, Fassone E, Tigano M, Lamperti C, Corti S, Bordoni A, Fortunato F, Nizzardo M et al. (2009) The mitochondrial disulfide relay system protein GFER is mutated in autosomal-recessive myopathy with cataract and combined respiratory-chain deficiency. Am J Hum Genet 84(5):594-604.

SEQscoring: a tool to facilitate the interpretation of data generated with next generation sequencing technologies


Katarina Truvé[1], Oscar Eriksson[1], Martin Norling[1], Maria Wilbe[1], Evan Mauceli[2], Kerstin Lindblad-Toh[2,3], Erik Bongcam-Rudloff[1]

[1]Department of Animal Breeding and Genetics, Swedish University of Agricultural Sciences, Uppsala, Sweden,
[2]Broad Institute of Harvard and Massachusetts Institute of Technology, Cambridge, USA,
[3]Science for Life Laboratory, Department of Medical Biochemistry and Microbiology, Uppsala University, Uppsala, Sweden

Depicted authors have names underlined.

Abstract

Next Generation Sequencing (NGS) technologies promise a revolution in genetic research. Generating enormous amounts of data, they bring both new opportunities and new challenges to researchers. SEQscoring was designed to facilitate analysis and enable extraction of the most essential information from data produced in NGS resequencing projects. Its main functionality is to help researchers locate the most likely causative mutations for a specific trait or disease, but it can advantageously be used whenever the goal is to compare and explore haplotype patterns, and to locate variations positioned in evolutionary conserved genomic elements. SEQscoring uses input data containing information about coverage and variations produced by other programs, like MAQ and SAMtools, and put the emphasis on methods for data visualisation and interpretation. We compare cases and controls in several ways and also utilise the power of comparative genomics, by scoring all variations according to their degree of conservation. SEQscoring is a publicly available, free, web-based service. It has an intuitive interface and can easily be used by biologists, medical researchers, veterinarians as well as bioinformaticians. We exemplify how SEQscoring was used in a recent study as a subsequent step to a genome- wide association study (GWAS) to extract a set of candidate mutations.

Availability: http://www.seqscoring.org

Introduction

"Next generation" sequencing (NGS) technologies are rapidly moving towards faster and cheaper resequencing of whole genomes and transcriptomes [1]. These new sequencing technologies promise to accelerate our knowledge of genetic variation and the associated phenotypic effects. As a consequence we might expect disease-causing mutations to be revealed and to see an advance in therapies and development of individually tailored drugs [2]. To be able to interpret the vast amounts of data being generated, new tools and algorithms will be needed for extensive comparison of entire individual genomes.

Resequencing not of entire genomes but of targeted regions has quickly become a valuable strategy to find candidate mutations following identification of associated regions using genome-wide association studies (GWAS). When performing GWAS, the use of SNP-chips, with thousands or several hundred thousands of single nucleotide polymorphisms (SNPs) evenly spread over the genome, makes it possible to locate disease-associated regions. Locations where allele frequencies differ between "cases and controls", may indicate a region harbouring a mutation where cases are identical by descent. Usually, a denser fine mapping of the located region(s) follows the GWAS. These methods have proven successful for identifying mutations inherited in a Mendelian fashion [3]. Most disease-causing mutations have been found in exons, probably because they have been subject to the most intense investigation, their causative effects being easier to validate than those of mutations in other regions. Yet, many regions outside exons have important regulatory effects, for example with respect to the location, timing and amount of gene expression. Particularly in complex diseases, where several genes and also environmental factors are involved, regulatory mutations are likely to be common. NGS allows the detection of variants in a wholly new scale. Consequently, we expect important mutations to be revealed with higher frequencies using these new methods, not just for those located in exons.

With new opportunities also come new challenges. The large amount of variation present in every individual (~1/1000 bp) raises the question of how to 'separate the wheat from the chaff'. The approach we outline here makes use of comparative genomics, as it has been shown

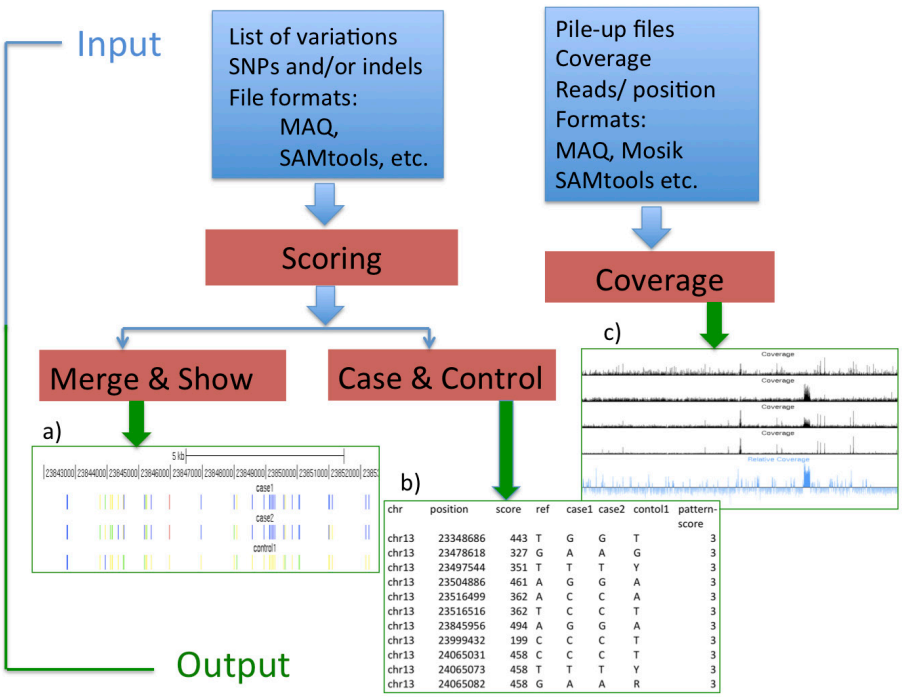


Figure 1. Overview of SEQscoring modules
The user submits input data in the form of lists of variants or coverage data produced by programs like e.g. MAQ or SAMtools. Variation data (SNPs/indels) are first scored by the Scoring module according to the degree of evolutionary conservation at the genomic position for the variation. In the next step data can be visualised using the Merge & Show module that aims to facilitate study of haplotype type structure and conservation within haplotypes. The Case & Control module, takes sample phenotype into account and aims to find the most likely positions or regions harbouring a causative mutation. The Coverage module performs calculations to find differences between cases and controls in an attempt to localize structural variants, like deletions or duplications. The output is provided in lists for download and further analyses and possibilities for direct visualisation in the UCSC browser. Some examples are shown in the figure; a) colour coded SNPs with a similar haplotype in cases, one SNP (red) within a conserved element; b) a table of conserved SNPs with calculated pattern-scores; c) visualisation of coverage differences between cases and controls.

that elements that are conserved across species, and are thus under purifying selection, are more likely to have a function [4-7]. We have therefore developed a tool, SEQscoring that scores mutations according to the degree of conservation, and also takes the pattern between cases and controls into account. The tool is freely available and can be accessed via the Web. The program also aims to facilitate the identification of structural variations, such as deletions and duplications, by calculating the ratio of average coverage between cases and controls in windows of a specified size. To allow comparison of individual datasets, some results are provided in a format compatible with the UCSC genome browser [8]. To facilitate the interpretation of data overall, SNPs and indels (small insertions or deletions) are colour coded in such a way as to give users an overview of features like homozygosity, conservation and variation. SEQscoring has been tested on several data sets, and shows great potential to help the user to extract the most essential information from their NGS-projects. The tool is easy to use and has an intuitive interface that can be used by biologists, medical researchers and bioinformaticians.

Results

Design and implementation

The SEQscoring tool aims to study haplotype structure and to localise important differences between cases and controls in genomic regions where NGS data are available. In Figure 1 we give an overview of the SEQscoring modules. The modules are described in more details below.

Prior to SEQscoring, variant detection should be performed using state of the art methods. Several different programs can be used to map millions of read to a reference sequence and to call variations, e.g. MAQ [9] and SAMtools [10]. SEQscoring supports several different file formats as input data and our ambition is to include additional formats if requested. Typically we expect

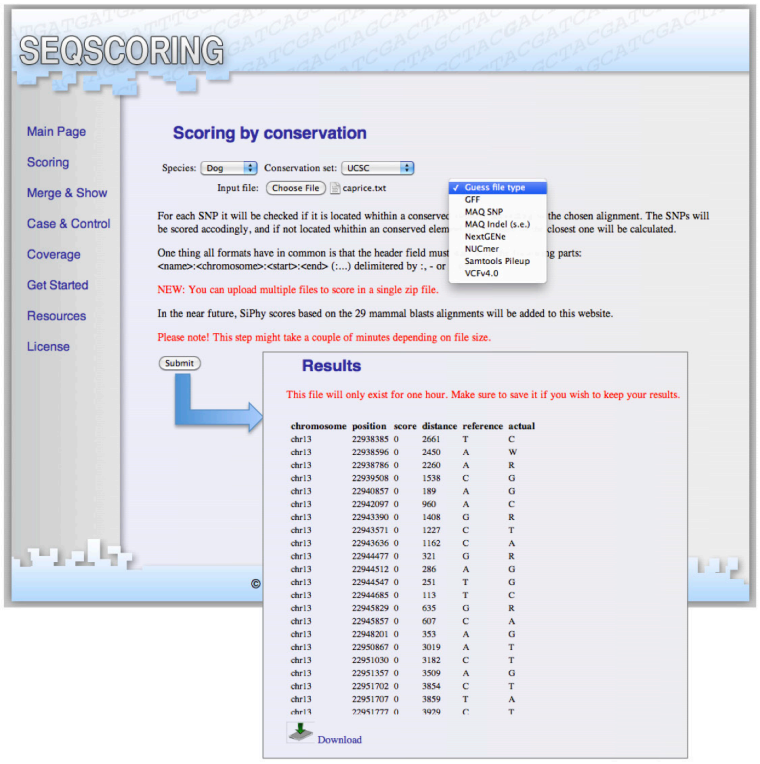


Figure 2. Conservation scoring identifies the variants with constraints and therefore with a higher chance to have a phenotypic effect. At the scoring page the users can submit their files of variations after choice of species and alignment/method for finding conserved elements. In the output file each variation has got a conservation score, and if not within a conserved element the distance to the closest one has been calculated.

the user to submit a single list with variants (SNPs and/or indels) for each individual.

To make this service accessible, it has been implemented as a web site hosted by an Apache web server running Python and Perl CGI scripts. Due to the dynamic content presented on the pages, the scripting language PHP is utilised for creating web pages. Python CGI scripts are used to catch both the raw data and the parameters from each form. Perl or Python modules then carry out the data processing. All data uploaded and produced by SEQscoring is stored for a limited period of time and then automatically erased. Submitted files get unique encrypted file names using MD5 sums in order to minimize the risk of access by unauthorized users.

Conservation scoring

In the Scoring module, variants are scored according to the degree of constraint at the genomic location for the variation. In principle, data from any species can be analysed as long as constraint score data is available for the particular species. For each variant the scoring module checks whether it is located within a constraint element, and if not the distance to the closest one is calculated. The location of constraint elements may differ depending on method and species used in the alignment. For mammals we propose the use of the 29mammals constraint scores (SiPhy omega or pi [11-12]) lifted onto the respective genome. Other available datasets are 16 amniota vertebrates and human/mouse/rat/dog comparison (Pecan [13] and PhastCons [14]). Those records are kept in our local database for high performance. Python modules performing iterative binary search have been implemented and compiled as C-extensions for fast and memory efficient conservation scoring of user submitted variations.

Visualization of variation and conservation

The Merge & Show module merges all variants and their score for all individual into a text file that can be downloaded for further analyses. The data is also displayed in the UCSC genome browser for easy comparison and investigation of haplotype structure. For easier interpretation SNPs

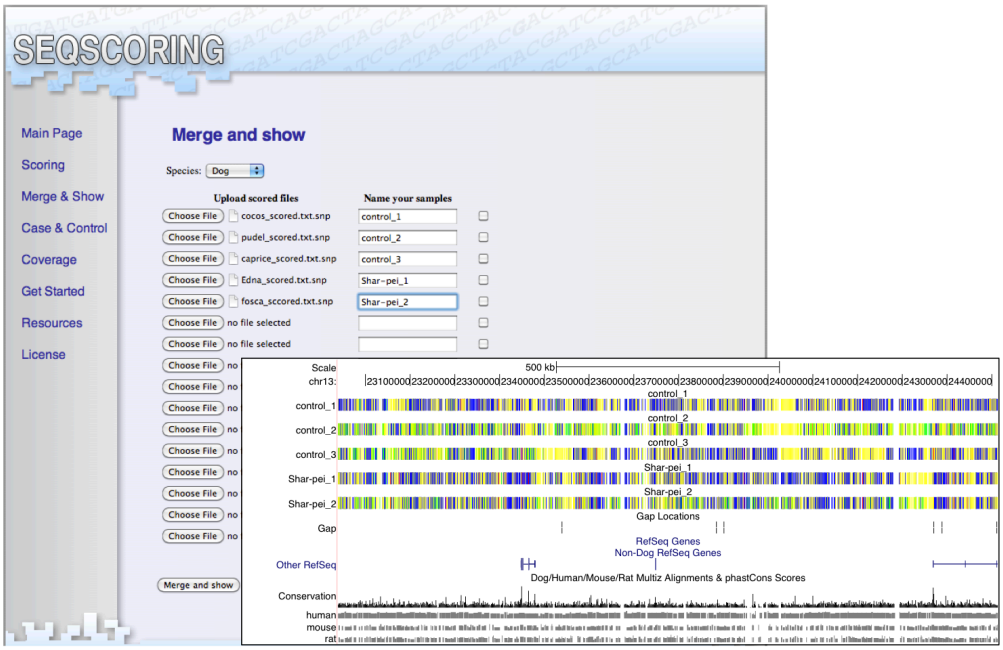


Figure 3. Merge samples and show variations in the UCSC Genome browser.
To ease the comparison of samples there is an option to merge and display scored files in the UCSC Genome Browser. Here an example of SNPs from five dog samples is displayed in the browser. SNPs are colour coded in the following way: yellow for homozygous equal to reference, blue for homozygous deviating from reference, green for heterozygous, red for homozygous within (± 5bp) a constraint element and pink for heterozygous within (± 5bp) a constraint element.

are displayed with the following colour code: homozygous SNPs within or near (± 5bp) constraint elements are coloured red; heterozygous SNPs within or near (± 5bp) constraint elements are coloured pink; non-constraint homozygous SNPs corresponding to the reference allele are coloured yellow; homozygous SNPs deviating from the reference are coloured blue; heterozygous non-constraint SNPs are coloured green.

Evaluation of concordance with phenotype status

The Case & Control module gives further help to reveal differences between cases and controls. Three different options are offered the user: 1) to compare constraint variants; 2) to compare genomic regions; 3) to transform data into a format for doing traditional association studies.

The first option selects SNPs located in conserved elements and scores them according to concordance with an expected pattern. The algorithm goes through all possible combinations of individual pairs and calculates a pattern-score depending on what the expected pattern of alleles are taking mode of inheritance into account. Cases and controls can be defined either based on phenotype or genotype expectation. We consider the highest scoring variants identified in this way to be among the most likely to be causative of the trait under investigation. Pattern-scores for conserved SNPs are calculated in the following way:

n = set of all samples
i= genotype for sample 1
j =genotype for sample 2
S(i) = status for sample (case or control)
p= pattern-score

$$p(SNP, I) = \left| \left\{ \begin{array}{c} (i,j) : i,j \in n \wedge \\ S(i) \neq S(j) \wedge i \neq j \vee \\ I = recessive \wedge S(i) = case \wedge S(j) = case \wedge i = j \vee \\ I = dominant \wedge S(i) = control \vee S(j) = control \wedge i = j \end{array} \right\} \right|$$

The second option "compare genomic region", scans for regions of specified size where cases are alike and differ from controls. Pair-wise combinations are examined in a similar way as for conserved SNPs, but all SNPs conserved and not conserved are taken into consideration. The mode of inheritance is not taken into consideration here. A sliding window approach is used and the highest score goes to the region that is as homozygous as possible in cases, and differ as much as possible to the controls. This option can be used to look for selective sweeps as well as for

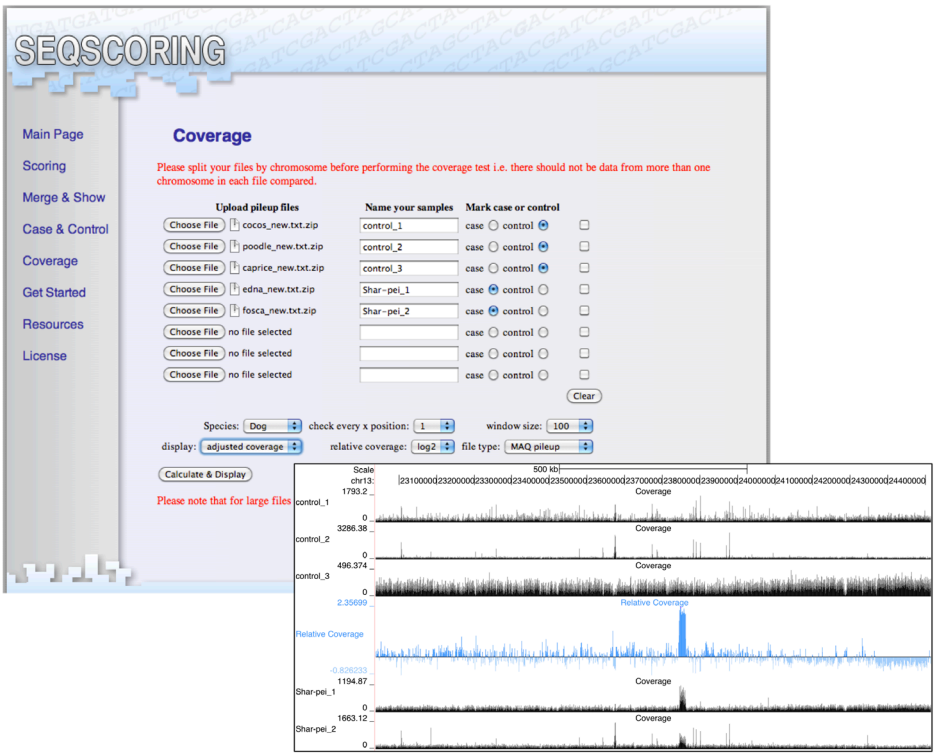


Figure 4. Comparison of coverage may reveal copy number variations.
The coverage option can be used to visualise differences in average coverage between cases and controls in an attempt to localize structural variations as duplications or deletions that might be causative for a certain phenotype. Here we show five samples from different dog breeds. Two of them are Shar-pei dogs with a thick wrinkled skin phenotype. The blue graph shows the coverage ratio (log2) between Shar-pei and the control breeds. Shar-peis clearly show a peak of excessive coverage that has now been proven to be a 16 kb duplication affecting both the skin phenotype and a fever disease in Shar-pei dogs.

smaller homozygous regions that might be identical by descent in cases harbouring a possible causative mutation for a specific trait.

Relative coverage analysis

The Coverage module is aimed to identify structural variation and at present it accepts pileup files created by MAQ [9], SAMtool [10] or Mosaik [15]. If there is a big difference in average coverage between data from different samples, the data can be normalised by setting the average coverage to the same fixed value for each individual. Comparable figures are thus calculated by dividing all data with an individual adjustment factor. There is also an option to average the coverage in a window of a specific size. The ratio of coverage between cases and controls is calculated for windows of a specified size and log2 transformed. The number of positions checked is limited to 150kb due to performance, thus giving a maximum resolution to regions smaller or equal to that size and subsequently diminishing resolution for larger regions.

Example

SEQscoring is currently in use at our lab for several resequencing projects where the aim is to find mutations responsible for specific traits or diseases in dogs. In our group, we traditionally use the dog as a disease model. The results obtained may, in many cases, be successfully translated to humans, and the knowledge gained thus has the potential to benefit both species [16-19]. It should be noted, however, that the methodology and software tool that we present here are generic, and not tied to a specific species.

In one of the first NGS projects at our lab the aim was to find the mutation responsible for the characteristic wrinkled skin phenotype in the Chinese Shar-pei dogs, a phenotype strongly selected for by breeders. The breed also suffers from a genetic disorder called Familial Shar-pei fever, a disease resembling human hereditary periodic fever syndromes. It has now been shown that the two features are connected and caused by a pleiotropic mutation [20]. We here exemplify the use of some SEQscoring functions with data

from the Shar-pei project. A region of 1.5 Mb had been selected for resequencing based on a genome wide SNP analyses showing strongly reduced heterozygosity in Shar-peis, implicating the presence of a selective sweep. The region was captured using custom designed arrays from NimbleGen and sequenced using Illumina Genome Analyzer. The obtained sequence reads were aligned to the target region of CanFam2.0 [21] using MAQ [9]. In Figure 2 it is illustrated how called SNPs are scored by conservation using SEQscoring. In this example we chose the UCSC phastCons alignment of four species. In the output file each variant has got a conservation score and, if not within a conserved element, the distance to the closest one has been calculated. In the first sequencing experiment two Shar-peis and three control breeds were sequenced. When the reads were mapped to a repeat masked reference ~1500 SNPs/individual were detected. In the next step we used the Merge & Show module. Downloading a text file with all SNPs/individual merged let us count that there were 3,430 SNPs in total and out of those only 84 were within conserved elements. The results are displayed in the UCSC genome browser (Figure 3) with colour coding as explained above. Next we used the Case & Control option to compare conserved SNPs, and found that only eight of the conserved SNPs had a pattern where the two Shar-peis were alike and differed from the controls. Those eight SNPs have been genotyped in several samples and in this case shown not to be causative since they were not unique for Shar-peis. Finally, we use the Coverage module to explore if there are any coverage differences between Shar-peis and controls. We targeted a region of 1.5 Mb and maximum of 150kb are displayed, meaning that in this case the program check the coverage at every 10th position. We also chose to use adjusted coverage and to average the coverage in a window size 100, actually meaning 100 *10 (every 10th position checked) = 1000 bases window. Coverage graphs were directly displayed in the UCSC genome browser. As can be seen (Figure 4) there was one clear peak of excessive coverage in both Shar-peis. The blue graph shows the log2 values of the ratio between cases and controls. It has now been shown that Shar-peis have a 16.1 kb duplication at this site [20].

Discussion

We have demonstrated how the use of the publicly accessible SEQscoring web site facilitates the interpretation of data from NGS-projects. We expect that the user, in most cases, is interested in localising the mutation for a specific phenotype. For best use of resources we propose a model where a number of individuals (6-12) are picked for resequencing, consisting of both cases and control. The region suspected to harbour the mutation has been narrowed down by GWAS before NGS.

It is assumed that genomic regions that are conserved across species are under evolutionary constraint and thus more likely to be functional. For this reason SEQscoring offers a fast conservation filtering of user submitted variations. It is important to be aware that different algorithms, and the set of species represented in the alignment, are likely to find different constraint elements. At present two different sets of constraint elements can be used for filtering. We are planning to add a third set in the near future, where conserved elements have been identified by alignment of 29 mammals using SiPhy [11]. In addition, a candidate function have been suggested for up to 60% of constrained bases [12]. We think that conservation filtering is an important and valuable step in variation evaluation but it should also be kept in mind that sometimes, functional elements show low degree of sequence conservation.

As mentioned in the results section there is an option to transform the NGS data to a format that can be used for traditional association studies based on allele frequencies using the program PLINK [22]. Usually a small number of samples are under investigation by resequencing, and the sample size is not appropriate for large-scale association. However, in the case that a larger sample size is utilised we offer a down load format that allows export of data into plink format. We offer two other methods to evaluate the concordance of genotype with phenotype: to compare conserved SNPs, and to compare genomic regions that have been designed with the purpose to extract as much information as possible using relatively few samples. The option to compare conserved SNPs uses both the power of conservation filtering and the identification of a pattern in concordance with an expected mode of inheritance thus capable of extracting the most likely causative SNPs for a specific trait.

The option to compare genomic region would most likely find homozygous regions containing two risk alleles (homozygosity), and is therefore most applicable to recessive traits and traits under selection. Dominant traits and complex risk factors are harder to identify. Usually cases and controls are defined based on phenotype, but as haplotype information from the GWAS or fine-mapping is typically used when picking samples for resequencing, to increase the odds of localising the causative mutation we recommend to use controls that are believed to be homozygous for an assumed healthy wild-type haplotype.

Sometimes structural aberrations like insertions, deletions or duplications are responsible for a specific trait. The possibilities to detect such differences are limited in resequencing projects. If there is an insertion in one of the individuals, those reads will simply not mapped to the reference. The read-length is often quite short (~30-100 bp) limiting the size of repetitive regions that can be read through, meaning that differences in size of microsatellites, presence of LINES and SINES etc.,can be hard to detect. We propose the use of tools that do de novo assembly to be able to capture putative insertions and deletions. After assembling larger contigs those could be mapped back to the reference and thus detecting insertions, but still the maximum detectable insertion size would be approximately the size of the read length.

The use of paired end reads offers a possibility to detect larger insertions, duplications and deletions but will not recognise smaller differences since those might be due to different shearing size. For single end reads, the information about coverage at each position has proven to be useful for identifying the putative locations of copy number variation or deletions that differs between cases and controls.

NGS projects are likely to identify a vast amount of variations between individuals and it is challenge to extract the ones that might be functional. We propose a methodology where the goal for the analyses is to extract a limited set of variations most likely to be causative for the trait under investigation, and to continue the analyses by genotyping in several cases and controls. We showed that the analyses offered by SEQscoring are straightforward and easy to understand, but powerful and time saving through the ability to extract important information, visualise the results and help the user propose a set of candidate mutations from the vast amount of data produced.

Acknowledgments

We thank the 29mammals consortium for allowing us access to the SiPhy mammalian constraint elements. This work has been supported by the EMBRACE project funded by the European Commission within its FP6 Programme, under the thematic area "Life sciences, genomics and biotechnology for health", contract number LHSG-CT-2004-512092 and Bioinformatics Infrastructure for Life Sciences (BILS) funded from the Swedish Research Council. KLT is funded by a EURYI from the ESF.

Competing interest statement

None declared

References

1. Shendure J, Ji H (2008) Next-generation DNA sequencing. Nature biotechnology 26: 1135-1145.
2. Hodges E, Xuan Z, Balija V, Kramer M, Molla MN, et al. (2007) Genome-wide in situ exon capture for selective resequencing. Nature genetics 39: 1522-1527.
3. Altshuler D, Daly MJ, Lander ES (2008) Genetic mapping in human disease. Science 322: 881-888.
4. Woolfe A, Goodson M, Goode DK, Snell P, McEwen GK, et al. (2005) Highly conserved non-coding sequences are associated with vertebrate development. PLoS biology 3: e7.
5. Drake JA, Bird C, Nemesh J, Thomas DJ, Newton-Cheh C, et al. (2006) Conserved noncoding sequences are selectively constrained and not mutation cold spots. Nature genetics 38: 223-227.
6. Margulies EH, Blanchette M, Haussler D, Green ED (2003) Identification and characterization of multi-species conserved sequences. Genome research 13: 2507-2518.
7. Birney E, Stamatoyannopoulos JA, Dutta A, Guigo R, Gingeras TR, et al. (2007) Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project. Nature 447: 799-816.
8. Fujita PA, Rhead B, Zweig AS, Hinrichs AS, Karolchik D, et al. (2011) The UCSC Genome

Browser database: update 2011. Nucleic acids research 39: D876-882.

9. Li H, Ruan J, Durbin R (2008) Mapping short DNA sequencing reads and calling variants using mapping quality scores. Genome research 18: 1851-1858.
10. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, et al. (2009) The Sequence Alignment/Map format and SAMtools. Bioinformatics 25: 2078-2079.
11. Garber M, Guttman M, Clamp M, Zody MC, Friedman N, et al. (2009) Identifying novel constrained elements by exploiting biased substitution patterns. Bioinformatics 25: i54-62.
12. Lindblad-Toh K GM, Zuk O, Lin M.F , Parker B.J (2011) A high-resolution map of evolutionary constraint in the human genome based on 29 eutherian mammals. Submitted.
13. Paten B, Herrero J, Beal K, Fitzgerald S, Birney E (2008) Enredo and Pecan: genome-wide mammalian consistency-based multiple alignment with paralogs. Genome research 18: 1814-1828.
14. Siepel A, Bejerano G, Pedersen JS, Hinrichs AS, Hou M, et al. (2005) Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes. Genome research 15: 1034-1050.
15. MOSAIK the reference-guided assembler: [http://bioinformatics.bc.edu/marthlab/Mosaik]
16. Karlsson EK, Lindblad-Toh K (2008) Leader of the pack: gene mapping in dogs and other model organisms. Nature reviews Genetics 9: 713-725.
17. Patterson DF, Pexieder T, Schnarr WR, Navratil T, Alaili R (1993) A single major-gene defect underlying cardiac conotruncal malformations interferes with myocardial growth during embryonic development: studies in the CTD line of keeshond dogs. American journal of human genetics 52: 388-397.
18. Mellersh CS, Boursnell ME, Pettitt L, Ryder EJ, Holmes NG, et al. (2006) Canine RPGRIP1 mutation establishes cone-rod dystrophy in miniature longhaired dachshunds as a homologue of human Leber congenital amaurosis. Genomics 88: 293-301.
19. Green SL, Tolwani RJ, Varma S, Quignon P, Galibert F, et al. (2002) Structure, chromosomal location, and analysis of the canine Cu/Zn superoxide dismutase (SOD1) gene. The Journal of heredity 93: 119-124.
20. Olsson M, Meadows JRS, Truvé K, Rosengren-Pielberg G, Puppo F, Mauceli E. (2011) A Novel Unstable Duplication upstreams of HAS2 predisposes to a Breed-defining skin Phenotype and a Periodic Fever Syndrome in Chinese Shar-Pei Dogs. PLoS Genet 7(3):e1001332.
21. Lindblad-Toh K, Wade CM, Mikkelsen TS, Karlsson EK, Jaffe DB, et al. (2005) Genome sequence, comparative analysis and haplotype structure of the domestic dog. Nature 438: 803-819.
22. Purcell S, Neale B, Todd-Brown K, Thomas L, Ferreira MA, et al. (2007) PLINK: a tool set for whole-genome association and population-based linkage analyses. American journal of human genetics 81: 559-575.

e-RGA: enhanced Reference Guided Assembly of Complex Genomes



Francesco Vezzi[1,2], Federica Cattonaro[2], Alberto Policriti[1,2]

[1]DIMI department of Informatics and Mathematics University of Udine, Udine, Italy, [2]IGA institute of applied genomics, Udine, Italy,

Abstract

Next Generation Sequencing has totally changed genomics: we are able to produce huge amounts of data at an incredibly low cost compared to Sanger sequencing. Despite this, some old problems have become even more difficult, *de novo* assembly being on top of this list. Despite efforts to design tools able to assemble, *de novo*, an organism sequenced with short reads, the results are still far from those achievable with long reads. In this paper, we propose a novel method that aims to improve *de novo* assembly in the presence of a closely related reference. The idea is to combine *de novo* and reference-guided assembly in order to obtain enhanced results.

Introduction

DNA sequencing is becoming cheaper every day. Instruments like the Illumina HiSeq 2000 or Solid 4 System are able to produce higher than 30X coverage of a human genome for less than $10,000. Next Generation Sequencing (NGS) allows sequencing at low cost and at a fraction of the time with respect to the Sanger Method [1]. NGS technologies (Illumina, Roche and Solid, to mention just a few) are capable of producing an enormous amount of raw data (for a complete review, see [2]). The throughput of such instruments is increasing so fast that descriptions of their performance became obsolete in just a few months.

Even though many papers have presented high-quality assemblies based on NGS data (see [3,4]), *de novo* assembly, especially for large-genome eukaryota, is still a 'holy grail' [5].

When a completely new organism is to be sequenced, the basic assembly strategy is still the celebrated Whole-Genome Shotgun (WGS) method. A number of tools that aim to perform *de novo* assembly using NGS data have been proposed. Among the most popular, we mention SOAPdenovo [6] and ABySS [7] (see [8] for an updated list). The assemblies produced by tools designed for NGS data are, in general, not comparable in quality with the assemblies produced by instruments like Arachne [9] and those designed for capillary sequencing data. The reason for this is that while, with NGS instruments, coverage is no more a bottleneck, read length, as well as the reduced size of the insertion between paired reads, makes correct assembly and positioning of repeats much more of an issue. *De novo* assembly with short reads is still very difficult [5] and, when assembling complex and repeated genomes, reasonably conservative *de novo* assembly programs are likely to produce collections of highly fragmented contigs. An interesting strategy to improve *de novo* assemblies has been termed "assembly reconciliation". The goal of assembly reconciliation is to merge the assemblies produced by different tools and to detect possible mis-assemblies [10].

The number of organisms whose genomes have been completely sequenced has been increasing rapidly each year and, for this reason, it is becoming viable to sequence an organism and then to align the sequence against a closely related genome. This strategy goes by the name of *Reference Guided Assembly* (RGA), its main advantage being the fact that, in general, even low coverage is sufficient to yield useful results. RGA consists of two phases: first, all the reads are aligned against the reference genome; then, a consensus sequence is extrapolated. Everywhere the coverage drops to zero, a sequence of Ns is placed. This problem has already been studied in the context of Sanger sequencing. In [11] and [12], two methods were proposed that use reference sequences to assist the assembly of new organisms. The challenge is becoming even more interesting with the advent of next generation sequencers, beginning with the technicalities and the practical considerations involved in the alignment phase.

As a matter of fact, many tools capable of rapidly aligning millions of short reads against a reference genome have been proposed recently. Tools like SOAP2 [13] and rNA [14] are essential for performing reference-guided assembly. The main problem with RGA and NGS is that (essentially for efficiency reasons) mapping al-

gorithms are highly conservative: it is possible to align reads with only a low number of errors and usually without gaps. In other words, we are able to reconstruct the conserved regions, while we cannot reconstruct areas that are divergent and (usually) more interesting. While, for example, there are techniques that use the paired-reads information to identify insertions/deletions (structural variations) [15,16], there is no clear way to reconstruct them.

In [17], a tool (MAIA) has been proposed to integrate multiple *de novo* and reference-guided assemblies. This tool uses the output of different assemblers, and of different reference-guided assemblies obtained with several reference sequences, to improve the final assembly result. MAIA constructs an overlap graph from the pairwise alignments of all the contigs. In large and repetitive genomes, like plant genomes, this step is computationally expensive and could easily lead to a large number of ambiguous or false overlaps.

Velvet's Columbus module tries to improve the assembly results using a reference sequence. In particular, the Columbus module aims to reconstruct candidate structural variations. Again, in this case, as a consequence of the repeats, this approach cannot be used on large, repetitive genomes.

In this paper, we briefly discuss and present results of a novel strategy to assemble genomes in the presence of a related sequence. In particular, we study how to merge the *de novo* and reference-guided assembly strategies, in order to assemble new organisms and improve the result achievable using only either one of the two. We will show how, by applying some of the ideas of assembly reconciliation, we can obtain an *enhanced reference assembly* of a new organism. All the alignments are guided by the reference sequence, in this way avoiding mis-assemblies and ambiguous overlaps. In [18], we illustrated the results obtained by our pipeline on a set of small organisms (chloroplasts and microbial); in this work, we show how the same pipeline can be effectively used for the assembly of large, highly repetitive and heterozygous genomes (plant genomes).

Reference-guided assembly

When a reference sequence *A* and a set of reads *R* are given, there are essentially two possible ways to perform reference assembly. The standard way simply involves aligning all the reads in *R* against the reference *A*, and then obtaining some consensus sequences. In the text that follows, we refer to this method as standard-RGA (*s-RGA*), and similarly, we call the consensus sequence produced *s-A*. An alternative approach is to perform de novo assembly on *R* first, and then to align the resulting contigs against the reference *A*. We call this second method de novo-RGA (*dn-RGA*), and the consensus sequence produced from it *dn-A*. Both the output sequences have "N" everywhere the coverage drops to 0. In order to simplify the discussion, we suppose *A* to be a single sequence (and therefore *s-A* and *dn-A* are also single sequences). It will be clear that this is not a limitation.

In the presence of NGS data, we have to use aligners like SOAP2 [13] and rNA [14] to obtain *s-A*. These aligners are highly conservative, they allow alignment of reads with a low number of errors, usually without gaps. For this reason, the length of *s-A* is the same of *A*. The sequence *dn-A* is obtained in three phases: first, reads are assembled using a short-read assembler ([6,7]); the resulting contigs are then aligned against *A*; after this, the consensus is generated. The challenge is to find an order for the contigs generated through the *de novo* assembly procedure. Several tools have been proposed to address this task. OSLay [19] computes a synthetic layout of the contigs using a reference sequence to anchor the *de novo* sequences; the Mauve aligner [20] gives as output an ordered version of the *de novo* contigs; and PGA [21] is able to layout the contigs with more than one reference genome at a time using global searches. All these tools implement or use a BLAST-like [22] search to align contigs against the reference. This alignment technique allows us to place reads on a reference with low similarity constraints. In particular, the contigs can be aligned against the reference sequence allowing partial hits and gaps.

This situation is similar to the already studied situation of assembly reconciliation. Casagrande *et al.* [10] proposed a method capable of merging two draft assemblies without performing global alignment. In particular, they proposed the use of one of the two assemblies as an "anchor", in order to resolve conflicts (*Master Assembly*).

Given the two assemblies *s-A* and *dn-A*, a suitable adaptation of this idea can be applied

in the current context to obtain an enhanced reference assembly.

Methods

The Merge Graph: Definition

In this section, we briefly sketch the formal steps necessary to define the Merge Graph, at the base of our technique. More details can be found in [18].

With *S[i,j]* (*S* being either Δ or Γ), we identify the so-called "slice" of a string *S*, namely the substring of S from position *i* to *j*. If *S[i,j]* belongs to {*a,c,g,t*}*, we call it a *(pure) contig*, while if *S[i,j]* belongs to {*N*}*, it is named *gap*. In this context, when a (pure) contig is maximally extended, we say that it is a *max-contig*, and we define, in an analogous way, a *max-gap* (in general, we speak of max-area). Given two strings δ and γ, the function *D(δ,γ,d)* returns a value between 0 and 1, representing a percentage difference between the two strings. This value is naturally computed using a distance metric d (e.g., Hamming). The merge graph *MG(Δ,Γ)* is a directed graph such that *V* is contained in $I_\Delta \times I_\Gamma$ (I_Δ and I_Γ being all possible intervals in Δ and Γ, respectively) and can be partitioned into four sets: *gap-nodes* (V_g, gap against gap), *delta-nodes* (V_δ, a Δ-contig against a Γ-gap), *gamma-nodes* (V_γ, a Δ-gap against a Γ-contig) and *merge-nodes* (V_m, contig against contig). We also fix *t* and *s*, two thresholds that express bounds on the distance (alignment similarity) and the absolute position (within the relative string), respectively.

Table 1. Table of symbols and definitions.

A	Reference sequence
s-A	Consensus sequence obtained aligning short reads against reference A
dn-A	Consensus sequences obtained aligning de novo contigs against reference A
e-A	Enhanced reference-guided assembly obtained through the e-RGA pipeline
Δ, Γ	Generic sequences
MG(Δ,Γ)	Merge graph for the sequences Δ and Γ
MGA	Merging Global Alignment

Edges are defined in such a way as to connect pairs of intervals (one on Δ and one on Γ) that can subsequently be put in the output as-

```
          0          11         21         31         41         51         61
Δ:  NNNNNNNaag gtttaaggtc ctacaNNNNa ctcatcataa aaacccNNNN NNNNNNctct aggtaaaa
Γ:  NNNNNNNNNN NNNNggccta cacNNNNNac tcatcatcaa aaacccNNaa aaaaNNNNNN NNNNaaaaa
```

Figure 1 (a). The strings Δ and Γ (|Δ|=68, |Γ|=69).

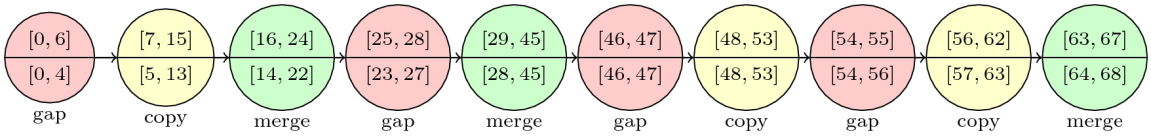


Figure 1 (b). A possible Merge Graph G for Δ and Γ with s=2 and t=2.

```
Δ′:  NNNNNNN AAGGTTTAA GGTCCTACA- NNNN- ACTCATCATAAAAA-CCC NN NNNNNN NN- CTCTAGG TAAAA
Γ′:  NNNNN-- NNNNNNNNN GG-CCTACAC NNNNN ACTCATCATCAAAAACCC NN AAAAAA NNN NNNNNNN AAAAA
Λ:   MMMMMII MMMMMMMMM MMIMMMMMMD MMMMD MMMMMMMMMSMMMMDMMM MM MMMMMM MMD MMMMMMM SMMMM
```

Figure 1 (c). A possible Global Alignment obtained from the Merge Graph G (M means match, S means substitution, I means insertion, while D means deletion).

```
Δ′:  NNNNNNN AAGGTT- TAAGGTCCTACA- NNNN- ACTCATCAT-AAAAACCC NN NNNNNN NN- CTCTAGG TAAAA
Γ′:  NNNNNNN NNNNNNN ---GG-CCTACAC NNNNN ACTCATCATCAAAAACCC NN AAAAAA NNN NNNNNNN AAAAA
                     constrain
                     violated
```

Figure 1 (d). A Global Alignment that does not allow the creation of an MGA.

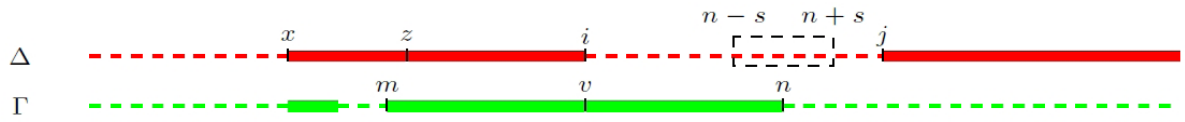


Figure 2. MGA construction. A possible scenario.

sembly. The resulting graph must be acyclic. In Figure 1B, an example of a merge graph is given.

In order to clarify the notation, we have summarised the symbols used throughout the paper in Table 1.

Merge Graph and Global Alignment

Given the strings Δ and Γ, there is a deep connection between the merge graph *MG(*Δ*,*Γ*)* and a global alignment between them. Building global alignments turns out to be equivalent to building a merge graph. In following text, we refer to the global alignment ensuing from a merge graph as *Merging Global Alignment (MGA)*.

The merge graph *MG(*Δ*,*Γ*)* can be used to extract a family of edit strings. From each node, we can produce an edited version of the two substrings that are represented. From delta and gamma nodes, we simply extract the contigs, while from the merge and gap nodes we can produce an edited version of one of the two strings. In the case of gap nodes, the edited version will contain only insertions and deletions (see Figures 1A, 1B and 1C). Once the edit strings are computed, the corresponding *MGA* can be calculated.

Given an *MGA*, the construction of a merge graph is more complicated. An *MGA* is a global alignment with two kinds of local properties: *locality* and *similarity*. In general, a global alignment does not guarantee these local properties, hence we can easily construct a global alignment that violates a local constraint. It can be proved that the global alignments we are seeking must respect the following properties: if Δ*[i,j]* and Γ*[k,l]* are aligned one against the other, then $|i-k|-1<s$ and $|j-l|-1<s$ (locality); if Δ*[i,j]* and Γ*[k,l]* are aligned and at least one is a max-contig or $\Delta[i-1]=\Gamma[l+1]=N$ (or $\Delta[j+1]=\Gamma[k-1]=N$), then *D(* Δ*[i,j]*,Γ*[k,l],d*) (similarity).

If such an alignment exists, calling Δ' and Γ' the two strings over the alphabet {*a,c,g,t,N,-*} returned by the global alignment between Δ and Γ, we can build *MG(*Δ*,*Γ*)* by simply reading from left to right Δ' and Γ'. For each position, we have to judge if a new node is beginning, or if we can continue extending the current one.

The determination of this global alignment can be computationally cumbersome. We cannot simply use an algorithm that calculates an optimal sequence alignment because a choice that can create an optimal global alignment will not necessarily lead to an alignment that respects all the local constraints (see Figure 1(d)). We will now sketch a complete algorithm that, given the strings Δ and Γ, generates all possible *MG(*Δ*,*Γ*)*.

The algorithm starts by reading the two sequences from left to right. For every contig in Δ and Γ, we can recursively compute all the possible alignments that satisfy the locality, and possibly the similarity, constraints. More detail is given in Figure 2. Let us assume that the last generated node was *<[z,i],[m,v]>*. From the merge graph definition, we have that at least one of i or v must be the end of a max-area, in this case i. At this point, we have to calculate the nearest (from *i*) max area end, *n* in the case shown in Figure 2. So the node we are going to create is *<[i,k],[v,n]>*, with $n-a<k<n+s$ (paying attention to some special case, we can reduce the search space). In order to generate all the possible graphs, we have to recursively generate all the nodes. In case we are generating a merge node, we have also to check if the similarity constraint is respected. The algorithm terminates because at every step it proceeds forward along both strings.

Minimal Merge Graph

Given two strings Δ and Γ, it is clear that the existence of an *MG(*Δ*,*Γ*)* depends on the two thresholds *s* (the bound on the relative distance between intervals involved in the same node) and *t* (the bound on the similarity distance between strings involved in the same merge node). By setting s to be large enough, we can easily go towards computing a merge graph composed only by gap, delta and gamma nodes. Another trivial solution can be found when *t* is set in such

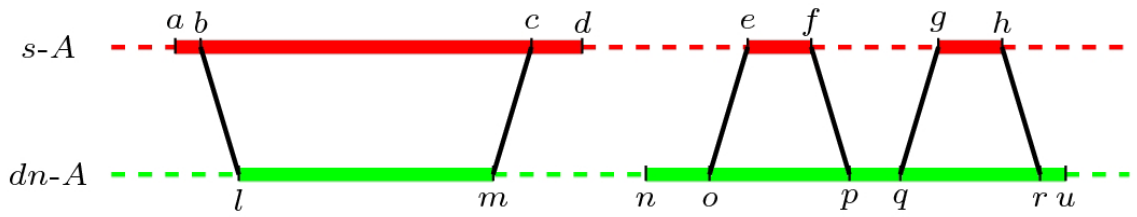


Figure 3. Practical identification of Merge-nodes for the Merge Graph Construction.

a way that merge nodes accept very low levels of similarity.

It is clear that a merge graph between a pair of sequences is interesting when *s* is a small constant when compared to Δ and Γ lengths and when *t* is sufficiently strict. These two constraints can strongly help in designing a better performing algorithm.

Merge graph and reference-guided assembly

A merge graph is a data-structure able to describe a global alignment between two strings, with further constraints on local alignment and similarity. This data-structure can be used both to describe the relations between two strings, and to extract a consensus.

When working with *s-A* and *dn-A*, we elect one of the two sequences to be the *Master Assembly (MA)*, that is the assembly we believe to be correct. In practice, in presence of a merge node, instead of calculating a consensus, we simply keep the sequence from the *MA*. Usually, even though two choices are possible, the *MA* will almost always be *dn-A*, with regions present in the sequenced organism and absent in the reference. If the merge graph *MG(s-A,dn-A)* is available, it can be used to extract a new assembly. Each node *p* in *MG(s-A,dn-A)* is characterised by two intervals, *[i,j]* and *[k,l]*. For *p=<[i,j],[k,l]>*, we extract the sequence *dn-A[k,l]*, if *p* is a gamma or a merge node, *s-A[i,j]*, if *p* is a delta-node, or the shortest between *s-A[i,j]* and *dn-A[k,l]*, if *p* is a gap node. This assembly is named *e-A* (*enhanced Reference Guided Assembly*).

The difficult part is the *MG(s-A,dn-A)* construction. The correct and complete algorithm presented in section 3.2 takes a time proportional to $O(s(j-i)^2)$ for each max-contig in the worst case. Additionally, the parameters *s* and *t* are unknown and, in general, there is no clear way to estimate them in advance, or to at least sensibly approximate them.

When working with *s-A* and *dn-A*, we have that the *MG(s-A,dn-A)* merge graph must exist for some *s* belonging to the set *{0,...,|dn-A|-|s-A|}*. This follows directly from the construction of the two strings. It is more difficult to limit *t*. A good working approximation is the percentage difference allowed in the *de novo* contig alignment.

The particular context provided by *s-A* and *dn-A* allows us to further improve the construction algorithm, concentrating only on a significant subset of all the global alignments associated to *MG(s-A,dn-A)*. Thanks to some intrinsic properties of *s-A* and *dn-A*, the brute-force algorithm can be improved, avoiding the generation of all possible global alignments. See Figure 3 for a graphical representation, and [18] for further details.

Enhanced-rga: implementation details

The pipeline represented in Figure 4 was implemented using several third-party tools and a set of Perl scripts implemented by the authors.

In order to construct *s-A*, first, a short-string aligner is used to align all the reads against the reference sequence, and a consensus is then extracted. In all cases in which a read is found in multiple occurrences, we randomly choose one of the alignments. We used the short-string aligner rNA [14] and the "pileup" command provided by samtools [23] to extract the consensus sequence.

dn-A was obtained by first performing *de novo* assembly with ABySS [7] and the CLC assembler Cell 3.0 [24]. Together with SOAPdenovo, [6] these are the only two assemblers able to assemble complex genomes using a reasonable amount of time and RAM memory. We noticed that, using contigs from both assemblies, the amount of genome reconstructed in *dn-A* greatly improves. The delicate phase of mapping contigs against

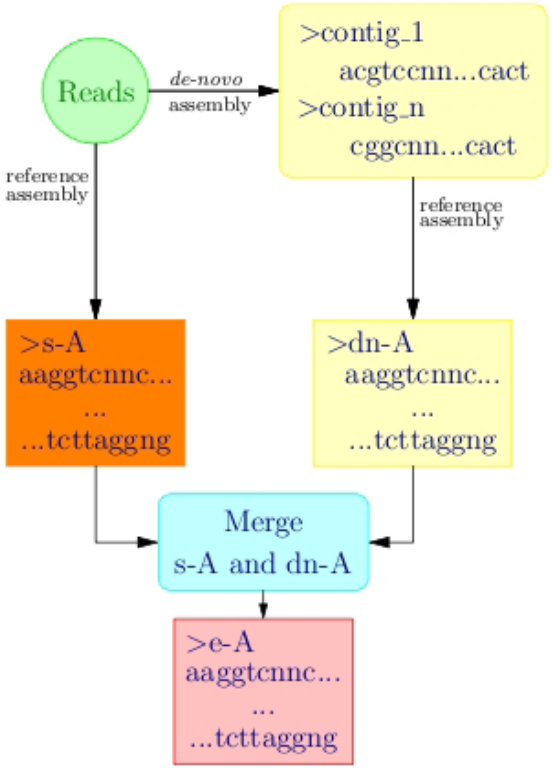


Figure 4. eRGA implementation

the reference sequence was accomplished with the CLC-Workbench[1].

Although we used these specific tools, clearly the production of *s-A* and *dn-A* can be carried out using different software without significant modification of the pipeline.

The core of *e-RGA* is the *MG(s-A,dn-A)* construction and *e-A* generation. These crucial phases are implemented within a Perl script that uses BLAST [22] to perform the approximate alignment. The program first memorises both *s-A* and *dn-A*, and localises all the *max-areas* (*max-contigs* and *max-gaps*). The *MG(s-A,dn-A)* construction proceeds as outlined in Section 4, with all the alignments performed by BLAST.

The software, together with a small example, can be downloaded at http://sole.dimi.uniud.it/~francesco.vezzi/software.php.

1 www.clcbio.com

Experiments and results

The Datasets

In [18], the *e-RGA* pipeline was tested on small genomes, demonstrating the effectiveness of our pipeline. We have further improved our pipeline and successfully used *e-RGA* on two large and complex plant genomes. The first dataset, named Sangiovese, comprises 5 Illumina lanes used to re-sequence a grapevine variety (Sangiovese, Rauscedo clone R24) with 100bp paired-end reads for a 90X total raw coverage. For this dataset, we used as reference sequence the genome of the highly homozygous grape clone, PN40024, used as reference genotype from the French-Italian Consortium for grape genome characterisation [25]. The second dataset, named Poplar, comprises 6 Illumina lanes used to sequence a Poplar individual belonging to the *Populus nigra* species, with 100bp paired-end reads for an 85X total raw coverage. In this case, we used as reference sequence the *Populus trichocarpa* genome [26]. While in the Sangiovese dataset we used a reference sequence belonging to the grapevine species, in the Poplar dataset the reference belongs to a different species. This difference is important in order to understand the differences obtained in the results. In both cases, before assembling and aligning, all the reads were filtered for quality, and we eliminated all sequences belonging to chloroplasts and mitochondria.

Both grapevine and poplar are characterised by long, repetitive genomes (480Mbp and 417Mbp respectively); moreover, both the sequenced individuals are highly heterozygous. These three conditions (length, repetitiveness and heterozygosity), together with the presence of two reference genomes, are perfect for our pipeline.

Results Discussion

Tables 2 and 3 summarise the results from the Sangiovese and Poplar datasets. As a measure of the assembly quality and correctness, we report the percentage of aligned reads (the same reads used to perform reference and *de novo* assembly), the number of contigs reconstructed, the mean contig length, the L50g (the length of the longest contig such that the sum of all the contigs greater than it represents half the expected genome length) and, in brackets, the L50c (the length of the longest contig such that the

sum of all the contigs greater than it represents half the total contig length), and the percentage of Ns in the sequence. The L50g gives us a normalised value that describes the connectivity level of the assembly.

Those statistics have been computed for the reference sequence A, for the *s-RGA* output *s-A*, for the *de novo* assembly output *dn*, for the *dn-RGA* output *dn-A*, and finally, for the *e-RGA* output *e-A*.

Statistics such as the mean contig length, contig number, L50c and L50g, give us an idea of the quality of the assembly. In the Sangiovese case, we can see how the mean length obtained through *e-RGA* is longer than the other approaches, and although the *dn-A* mean length has a close value, we must consider the fact that these contigs cover only half the genome length as described by the high percentage of unknown characters. In the Sangiovese dataset, the most impressive results are the L50g and L50c improvements. Both *e-A*'s L50g and L50c are better than those of *s-A*, and they largely improve the results achievable with *de novo* assembly alone. This shows that our pipeline can effectively improve the final assembly result.

Similar results are summarised in Table 2 for the Poplar dataset. Owing to the distance between the sequenced organism (*Populus nigra*) and the reference genome (*Populus trichocarpa*), the Poplar results can look less promising than those of Sangiovese. However, the number of mapped reads against *e-A* is higher than the number of reads mapped against both *s-A* and *dn-A*. The fact that we are able to map a higher number of reads against *dn* should also be a consequence of the distance between the reference and the sequenced genome. As far as the standard assembly statistics are concerned (L50g, L50c and mean contig length), we can again see how the *e-A* results are better than those achievable by simply mapping reads or contigs back to the reference. Despite the *de novo* assembly result looking much better than the other approaches, we must stress the fact that *de novo* assembly alone gives us a set of 289,854 unordered contigs, with no information about their position in the final genome. It would be interesting, but outside the scope of this work, to understand the composition of the contigs not used to construct *dn-A*. These contigs, if correctly assembled, represent the areas belonging to the sequenced organism exclusively.

A further measure of the improvements introduced by the use of *e-RGA* is the number of successfully aligned paired reads (*i.e.*, paired reads

Table 2. Results obtained for the Sangiovese dataset.
For all the techniques used, we show the percentage of aligned reads, the number of contigs, the mean contig length, the L50 length computed both on the expected genome length and on the total contig length, and the number of unknown characters "N".

	Sangiovese				
	% aligned reads	Contigs number	Mean contig length	L50g (L50c)	% Ns
A	80.21%	-	-	-	3.00%
s-A	80.99%	246752	1758 bp	8514 bp (9901 bp)	7.64%
dn	53.10%	289854	1942 bp	1753 bp (3328 bp)	0.70%
dn-A	50.71%	109833	2246 bp	600 bp (3947 bp)	47.70%
e-A	81.77%	198194	2282 bp	12494 bp (14219 bp)	6.40%

Table 2. Results obtained for the Poplar dataset.
For all the techniques used, we show the percentage of aligned reads, the number of contigs, the mean contig length, the L50 length computed both on the expected genome length and on the total contig length, and the number of unknown characters "N".

	Poplar				
	% aligned reads	Contigs number	Mean contig length	L50g (L50c)	% Ns
A	55.00%	-	-	-	2.14%
s-A	58.00%	778065	365 bp	525 bp (1105 bp)	25.22%
dn	67.84%	116683	2728 bp	2906 bp (4487 bp)	0.40%
dn-A	37.00%	77370	1335 bp	0 bp (2085 bp)	62.46%
e-A	59.00%	558762	482 bp	957 bp (1959 bp)	18.56%

that align on the sequence at the expected distance and orientation). In both datasets, *e-A* is the sequence on which the largest number of constraints is respected.

More Applications

A possible *e-RGA* application, not explored in this work, is the identification and, more importantly, the reconstruction of structural variation. Two different steps of the *e-RGA* pipeline can be instrumental to this purpose. First is in the construction of *dn-A*. We can identify contigs that are aligned with gaps: alignments that introduce gaps in the reference sequence represent a putative insertion in the sequenced genome; conversely, alignments that introduce gaps in contigs reveal a putative deletion in the sequenced genome. The second step in which we might be able to identify structural variations is in *e-A* construction from the *MG(s-A,dn-A)* graph. In this case, a gamma-node *<[i,j],[k,l]>* (*dn-A*-contig against a *s-A*-gap) in which the interval *[i,j]* (the *s-A*-gap) is shorter than *[k,l]* (the *dn-A*-contig) is witness to an insertion in the sequenced genome; conversely, if the interval *[i,j]* is longer than *[k,l]* then the node is witness to a deletion in the sequenced genome. In the case of *delta-nodes*, the situation is symmetric.

Conclusions

The *e-RGA* pipeline was successfully applied to small organisms in [18]; in this paper, we have shown how the same approach can easily scale to large datasets with high coverage over complex (large and highly repetitive) genomes like the Grapevine and Poplar genomes.

e-RGA needs a reference genome belonging to a closely related organism. With the number of available genomes growing at a speed believed impossible only few years ago, this requirement is becoming standard.

Several research efforts are ongoing to design tools to identify and study *structural variations* (SV) [16], in particular within individual genomes. One major stumbling block is that it is still unclear how the identified or putative SV can be reconstructed. A future development of *e-RGA* will be to output putative SV identified during *e-A* construction. In this way, our pipeline, coupled with a tool able to identify/verify SV, could be used to reconstruct sequences that are specific to a particular individual.

Competing interest statement

None declared

References

1. Metzker ML (2009) Sequencing technologies — the next generation. Nat Rev Genet 11: 31-46.
2. Mardis ER (2008) The impact of next-generation sequencing technology on genetics. Trends Genet: 24: 133-141.
3. Li R, Fan W, Tian G et al. (2009) The sequence and de novo assembly of the giant panda genome. Nature 463: 311-317.
4. Velasco R, Zharkikh A, Affourtit J et al. (2010) The genome of the domesticated apple (Malus × domestica Borkh.). Nat Genet: 42: 833-839.
5. Nagarajan N, Pop M (2009) Parametric complexity of sequence assembly: Theory and applications to next generation sequencing. J Comput Biol 16: 897-908.
6. Li R, Zhu H, Ruan J al. (2010) De novo assembly of human genomes with massively parallel short read sequencing. Genome Res 20: 265-2727.
7. Simpson J, Wong K, Jackman S et al. (2009) ABySS: A parallel assembler for short read sequence data. Genome Res 19: 1117-1123.
8. Miller J, Koren S, Sutton G (2010) Assembly algorithms for next-generation sequencing data. Genomics 95(6): 315-327.
9. Batzoglou S, Jaffe DB, Stanley K et al. (2002) ARACHNE: A Whole-Genome Shotgun Assembler. Genome Res 12: 1100-1105.
10. Casagrande A, Del Fabbro C, Scalabrin S et al. (2009) GAM: Genomic Assemblies Merger: A Graph Based Method to Integrate Different Assemblies. Proc IEEE International Conference on Bioinformatics and Biomedicine (BIBM) 321-326.
11. Pop M, Phillippy A, Delcher AL et al. (2004) Comparative genome assembly. Brief Bioinform 5:237-48
12. Gnerre S, Lander ES, Lindblad-Toh K et al. (2009) Assisted assembly: how to improve a de novo genome assembly by using related species. Genome Biol 10: R88.
13. Li R, Yu C, Li Y et al. (2009) SOAP2: an improved ultrafast tool for short read alignment. Bioinformatics 25: 1966-1967.

14. Policriti A, Tomescu A, Vezzi F (2010) A Randomized Numerical Aligner (rNA). Proc Language and Automata Theory and Applications (LATA) 6031: 512–523.
15. LeeS, Hormozdiari F, Alkan C et al. (2009) MoDIL: detecting small indels from clone-end sequencing with mixtures of distributions. Nat Methods 6: 473–474.
16. Kerstens HH, Crooijmans RP, Dibbits BW et al. (2011) Structural variation in the chicken genome identified by paired-end next-generation DNA sequencing of reduced representation libraries. BMC Genomics 12: 94-110.
17. Nijkamp J, Winterbach W, Broek M et al. (2010) Integrating genome assemblies with MAIA. Bioinformatics 26: i433-i439.
18. Cattonaro F, Policriti A, Vezzi F (2010) Enhanced reference guided assembly. Proc IEEE International Conference on Bioinformatics and Biomedicine (BIBM) 77-80.
19. Richter DC, Schuster SC, Huson DH (2007) OSLay: optimal syntenic layout of unfinished assemblies. Bioinformatics 23: 1573-1579.
20. Rissman AI, Mau B, Biehl BS et al. (2009) Reordering contigs of draft genomes using the Mauve aligner. Bioinformatics 25: 2071-2073.
21. Zhao F, Zhao F, Li T et al. (2008) A new pheromone trail-based genetic algorithm for comparative genome assembly. Nucleic Acids Res. 36: 3455-3462.
22. Altschul S. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. 25: 3389-3402.
23. Li H, Handsaker B, Wysoker A et al. (2009) The Sequence Alignment/Map format and SAMtools. Bioinformatics 25: 2078-2079.
24. CLC Team (2010) De novo assembly on the CLC Assembly Cell. (White Paper) http://www.clcbio.com/white-paper/
25. Jaillon O, Aury JM, Noel B et al. (2007) The grapevine genome sequence suggests ancestral hexaploidization in major angiosperm phyla. Nature 449: 463-467.
26. Tuskan GA, Difazio S, Jansson S et al. (2006) The genome of black cottonwood, Populus trichocarpa (Torr. & Gray). Science 313: 1596-1604.

do it yourself

Vivienne Baillie Gerritsen

It can take ages to meet the right partner. So much so that plants lost their patience millions of years ago and thought up something else: the art of selfing. Many flowering plants are indeed capable of extensive in-breeding – by way of a rather subtle form of hermaphroditism – to ensure their spread and survival. The common mouse-ear cress, *Arabidopsis thaliana*, which has become the model plant for botanists, is revealing how many plants are able to perpetuate their species by letting their pollen fertilise their own pistil. Which prompts the question: how does any given plant species avoid self-fertilisation in the first place? The answer, or at least part of it, is: the S locus. The S locus carries two genes whose protein products – SCR and SRK – are directly involved in *A.thaliana*'s capacity to self-pollinate or not, and may well illustrate the pathway used by many other plants.

"Hermafrodite", by Lisbeth Hummel

Courtesy of the artist

The notion that plants are able to self-pollinate is not new. Charles Darwin, who is widely known for his thoughts on the evolution of primates, also spent a lot of time making observations on countless other organisms, including plants. He had already suggested the existence of self-fertilisation during the second half of the 19th century – and even dedicated a book to the notion, *The Effects of Cross and Self Fertilisation in the Vegetable Kingdom (1876)*.

Why would plants seek to self-pollinate? The answer seems obvious. When you can't find someone else to do it, do it yourself. It is not in the pursuit of pleasure that a plant would end up self-pollinating but rather in the hope of guaranteeing successful growth within a given environment; an environment that has become hostile enough to make cross-breeding difficult, yet in which the disadvantages of in-breeding are outweighed by the advantages…

In the past few years, scientists have been studying one particular system in *Arabidopsis thaliana*, which has proved to be an essential part of an elegant process that has been coined "self-incompatibility" (SI). SI is used by plants to avoid self-pollination, thus ensuring genetic variation and population vigour. In a nutshell, plants whose SI system is working properly are not able to self-pollinate. It sounds straightforward enough, yet the SI system is turning out to be a complex one. However, what has been termed the S locus seems to play a pivotal role and is best illustrated by its protein products: s-locus receptor kinase (SRK) and s-locus cysteine-rich protein (SCR).

SRK is a transmembrane receptor protein, which probably forms a homodimer and is found on the very tip of a flower's pistil, known as its stigma. The stigma forms a kind of platform on which pollen is able to land, hydrate, germinate, and ultimately push its pollen tube all the way down the pistil to the

ovary to complete fertilisation. SCR is found in the pollen's outer coat and is secreted when pollen approaches the stigma. If the pollen and the pistil belong to the same plant, SCR and SRK belong to the same S locus and, like glove in hand, SCR will bind to SRK. Their binding then triggers off an alert system – probably via SRK phosphorylation – which interrupts pollen tube growth. As such, the system functions much like passing through customs – if the pollen belongs to the same plant, an alarm goes off and fertilisation is immediately stopped. It has been suggested that this happens by the degradation of actin filaments which support pollen tube growth.

This is how SI works for many plants. In *A.thaliana*, however, due to numerous mutations over time, the S locus has been out of order for about half a million years. As a consequence, when *A.thaliana*'s pollen settles on the tip of its own pistils, SCR is not recognised by the plant's SRK. Consequently, the SI alert is not set off and the pollen is left to germinate and make its way down the pistil to the ovary.

As always, no given pathway can be trimmed down to the likes of one or two proteins. Indeed, though a key element in plant self-incompatibility, the S locus is not the only decision-making entity in the SI pathway. A cascade of decision events occurs downstream and many other processes – such as pollen/stigma recognition, pollen hydration, germination and directional growth – upstream. Nevertheless, *A.thaliana* is turning out to be an excellent plant model for studying signalling pathways.

From a purely biological point of view, in-breeding – not to mention incest – has never been encouraged within a species, mainly for its healthy survival. Here's a thought: is it not amazing that the plant kingdom has resorted to a genetic system to discourage in-breeding, while humans count on words and cultural heritage?

Cross-references to UniProt

Defensin-like protein A, SCRA, *Arabidopsis thaliana* (Mouse-ear cress) : P0CG07
S-receptor-like serine/threonine-protein kinase SRK, *Arabidopsis thaliana* (Mouse-ear cress) : B0F2A9

References

1. Ivanov R., Fobis-Loisy I., Gaude T.
When no means no: guide to Brassicaceae self-incompatibility
Trends in Plant Science 15:387-394(2010)
PMID: 20621670

2. Tsuchimatsu T., Suwabe K., Shimizu-Inatsugi R., Isokawa S., Pavlidis P., Staedler T., Suzki G., Takayama S., Watanabe M., Shimizu K.K.
Evolution of self-compatibility in Arabidopsis by a mutation in the male specificity gene
Nature 464:1342-1347(2010)
PMID: 20400945

3. Boggs N.A., Nasrallah J.B., Nasrallah M.E.
Independent S-locus mutations caused self-fertility in Arabidopsis thaliana
PLoS Genetics. Volume 5, issue 3
PMID: 19300485

Protein Spotlight (ISSN 1424-4721), http://www.proteinspotlight.org, is published by the Swiss-Prot group at the Swiss Institute of Bioinformatics (SIB).

National Nodes

Argentina
IBBM, Facultad de Cs. Exactas, Universidad Nacional de La Plata

Australia
RMC Gunn Building B19, University of Sydney, Sydney

Belgium
BEN ULB Campus Plaine CP 257, Brussels

Brazil
Lab. Nacional de Computação Científica, Lab. de Bioinformática, Petrópolis, Rio de Janeiro

Chile
Centre for Biochemical Engineering and Biotechnology (CIByB). University of Chile, Santiago

China
Centre of Bioinformatics, Peking University, Beijing

Colombia
Instituto de Biotecnología, Universidad Nacional de Colombia, Edificio Manuel Ancizar, Bogota

Costa Rica
University of Costa Rica (UCR), School of Medicine, Department of Pharmacology and ClinicToxicology, San Jose

Cuba
Centro de Ingeniería Genética y Biotecnolgía, La Habana

Finland
CSC, Espoo

France
ReNaBi, French bioinformatics platforms network

Greece
Biomedical Research Foundation of the Academy of Athens, Athens

Hungary
Agricultural Biotechnology Center, Godollo

India
Centre for DNA Fingerprinting and Diagnostics (CDFD), Hyderabad

Italy
CNR - Institute for Biomedical Technologies, Bioinformatics and Genomic Group, Bari

Mexico
Nodo Nacional de Bioinformática, EMBnet México, Centro de Ciencias Genómicas, UNAM, Cuernavaca, Morelos

The Netherlands
Dept. of Genome Informatics, Wageningen UR

Norway
The Norwegian EMBnet Node, The Biotechnology Centre of Oslo

Pakistan
COMSATS Institute of Information Technology, Chak Shahzaad, Islamabad

Poland
Institute of Biochemistry and Biophysics, Polish Academy of Sciences, Warszawa

Portugal
Instituto Gulbenkian de Ciencia, Centro Portugues de Bioinformatica, Oeiras

Russia
Biocomputing Group, Belozersky Institute, Moscow

Slovakia
Institute of Molecular Biology, Slovak Academy of Science, Bratislava

South Africa
SANBI, University of the Western Cape, Bellville

Spain
EMBnet/CNB, Centro Nacional de Biotecnología, Madrid

Sri Lanka
Institute of Biochemistry, Molecular Biology and Biotechnology, University of Colombo, Colombo

Sweden
Uppsala Biomedical Centre, Computing Department, Uppsala

Switzerland
Swiss Institute of Bioinformatics, Lausanne

Specialist- and Assoc. Nodes

CASPUR
Rome, Italy

EBI
EBI Embl Outstation, Hinxton, Cambridge, UK

Nile University
Giza, Egypt

ETI
Amsterdam, The Netherlands

IHCP
Institute of Health and Consumer Protection, Ispra. Italy

ILRI/BECA
International Livestock Research Institute, Nairobi, Kenya

MIPS
Muenchen, Germany

UMBER
Faculty of Life Sciences, The University of Manchester, UK

CPGR
Centre for Proteomic and Genomic Research, Cape Town, South Africa

The New South Wales Systems Biology Initiative
Sydney, Australia

for more information visit our Web site
www.EMBnet.org

EMBnet.journal

ISSN 1023-4144

Dear reader,

If you have any comments or suggestions regarding this journal we would be very glad to hear from you. If you have a tip you feel we can publish then please let us know. Before submitting your contribution read the "Instructions for authors" at http://journal.EMBnet.org/index.php/EMBnetnews/about and send your manuscript and supplementary files using our on-line submission system at http://journal.EMBnet.org/index.php/EMBnetnews/about/submissions#onlineSubmissions.

Past issues are available as PDF files from the Web site:
http://journal.EMBnet.org/index.php/EMBnetnews/issue/archive

Publisher:
EMBnet Stichting
c/o Erik Bongcam-Rudloff
Uppsala Biomedical Centre
The Linnaeus Centre for Bioinformatics, SLU/UU
Box 570 S-751 23 Uppsala, Sweden
Email: erik.bongcam@bmc.uu.se
Tel: +46-18-4716696