Università degli Studi del Piemonte Orientale
Amedeo Avogadro

Dipartimento di Scienze e Innovazione Tecnologica

Dottorato di Ricerca in Chemistry&Biology
XXIX ciclo (A.A. 2013-2016)

Curriculum: Energy, environmental and food sciences
SSD: BIO/11

NGS DATA ANALYSIS APPROACHES FOR CLINICAL APPLICATIONS

Igor Saggese

Supervised by Prof. Flavio Mignone

Ph.D program co-ordinator: Prof. Domenico Osella

Contents

1 Introduction **5**

1.1 Molecular diagnostics 6

1.2 Denovo assembly of RNA-seq data 9

1.3 Application of denovo assembly to metagenomics . . . 10

2 Outline of the Thesis **17**

3 Analysis workflows for NGS **23**

3.1 A new pipeline for variant calling and annotation . . . 25

3.1.1 Analysis profiles 25

3.1.2 Alignment to reference 26

3.1.3 Sequencing error detection 28

3.1.4 Variant annotation 29

3.1.5 Integration with external resources 30

3.2 Amplicon Suite 31

3.2.1 Clinical validation 34

3.3 SmartVir 42

3.3.1 Results 44

4 Algorithms for NGS **53**

4.1 STAble 55

4.1.1 Head-tail alignments detection 56

4.1.2 Graph construction and traversal 66

4.1.3 Results post-processing 71
4.1.4 Parallel workflow 71
4.2 Benchmarks 72
4.2.1 Benchmarks on simulated datasets 72
4.2.2 Benchmarks on real data 78

5 Discussion **83**
5.1 Acknowledgements 86

6 List of Publications **89**
6.1 Journals and book chapters 89
6.2 Participation in conferences 89

Chapter 1

Introduction

Next generation sequencing (NGS) techniques are seeing constant evolutions and allow us to sequence nucleic acids with increasingly competitive cost per base and unprecedented speed: in just few hours it is possible to produce the same amount of data obtained with traditional techniques in the past 30 years [1].
The advent of these technologies answered the demand for greater and more advanced instruments to answer complex biological questions, allowing researchers and clinicians to probe genomes in greater depth.
Sanger sequencing [2], despite the impressive improvements since its introduction is not suitable for large-scale modern projects due to its high costs and low throughput.
NGS scenario is under constant evolution and many different platforms are available, such as *Illumina*, *IonTorrent* and *Roche 454* (that is currently at phase-out stage), however in every case the output is a vast set of reads (up to billions), each one representing a fragment of the input sample (typically in the range of 50-500 nucleotides) that have to be processed in a way that depends on the kind of analysis in order to get some meaning from them.
Analysis task is complicated by the presence of *sequencing errors*: although provided accuracy is usually very high (in the range of 98% to

99,9%) each platform has its specific error model that must be considered when performing analysis.
This problem is compensated by *sequencing depth*, that represents how many times each nucleotide from input sample has been sequenced: it is very unlikely to get the same errors in all reads, so by applying proper statistics it is possible to detect them. Adequate coverage is critical for accurate reassembly of the genomic sequence and to date it is not a problem to achieve very high depths.
Because of high throughput a NGS run on a single sample is likely to generate more data than is required, so with adequate protocols it is possible to multiplex multiple samples and to process them in a single run.
Regardless of the origin of the biological sample, NGS platforms produce very large files in which the reads from input sequences are coded and it may be not trivial at all to manage them and perform all the analysis steps required to extract useful information: bioinformatics analysis has become an essential part of the process.
The field of application for NGS techniques is wide and there is no "universal" way of analysing data. Depending on requirements of the specific project the need to adapt existing tools and pipelines or to develop new ones might arise.

The main aim of this research project was the development of new bioinformatic algorithms and integrated tools to address several aspects of NGS data analysis, with particular focus on three main areas: *molecular diagnostics* [3], *denovo assembly of RNA-seq data* [4] and application of denovo assembly to *metagenomics* [5].

1.1 Molecular diagnostics

Molecular diagnostics is a collection of techniques that allow to assess a person's health at molecular level by detecting specific markers in

his DNA and RNA and their possible effects on expressed proteins.
By genome sequencing it is possible to diagnose susceptibility to specific diseases and to get additional useful informations about patients on whom clinical actions must be taken.
This name identifies a wide set of techniques, however this project was mainly focused on the *identification of variants in gene panels* and *virus genotyping*.
A typical approach in this area is *target based amplification*, that consists in the design of specific primer pairs to amplify genomic regions where interesting markers are located: these regions are named *amplicons*.
Sequencing results are processed by a *variant calling* procedure, that consists on alignment of reads to a reference sequence to identify potential variants that are finally annotated with functional informations.
These applications are opening the doors to personalised medicine, however there is still a number of open challenges that must be addressed before they can be transferred to routine clinical practices, especially on the bioinformatics analysis side. The presence of sequencing errors and the large size of datasets are probably the most limiting factors that must be dealt with.

BRCA1 and BRCA2 (BRCA) genes are among the most frequently analysed genes in clinical routine, since rapid identification of germline BRCA mutations can be useful for both prophylactic strategies and therapy administration. In fact they are the two main highly penetrant genes predisposing to hereditary breast and ovarian cancer syndrome (HBOCS): about 5–10% of tumour cases are mainly caused by mutations in the BRCA tumour-suppressor genes, resulting in nonfunctional BRCA proteins. This defect compromises the accurate DNA repair function, cell cycle regulation and transcriptional activity [6, 7].
Sanger sequencing is considered the gold standard for identifying qualitative changes in BRCA regions: nevertheless, it is time consuming

and expensive, due to the large sizes of the coding regions of both genes (5592 and 10257 bp for BRCA1 and BRCA2 genes, respectively), and the equal distribution of mutations within regions of interest (Breast Cancer Information Core database [8] reports about 1781 and 2000 variants for BRCA1 and BRCA2 genes, respectively).
Introduction of NGS in laboratory practice allowed molecular diagnostic laboratories to increase the throughput and to analyse multiple genes in the same run, facilitating the study of complex disease where Sanger sequencing is not technically or economically feasible. However given current limitations these techniques are only applied as a support to traditional ones even though, with proper advancements, they are expected to replace them in the future.

Another important - and very current - molecular diagnostics application plays an important role in the treatment of HCV infected patients. Obtaining the correct genotype and subtype information about the infecting strain is essential to ensure that the most appropriate treatment regimen is selected. In addition once the viral population has been profiled it is also important to analyse mutations present on viral genome in order to detect those associated with drug resistances [9].
NGS has given a boost to this research area, since information provided by short reads can be used to unambiguously detect correct genotype [10] and genomic mutations. High sequencing depths also provide enough information to assess the presence of a mixed infection.

Our contribution to this area was the the development of a robust procedure for variant calling and annotation from NGS data specifically designed for clinical purposes that at present is used as core for two integrated tools: **Amplicon Suite** - for the detection and annotation of disease-related variants - and **SmartVir** - for virus genotyping and drug resistance detection.

The two applications and obtained results will be presented in chapter 3.

1.2 Denovo assembly of RNA-seq data

RNA-seq (also called *Whole-Transcriptome shotgun sequencing (WTSS)* [11]) is a set of powerful techniques that allow to perform identification and quantification of RNA transcripts in a biological sample.
Unfortunately given current limitations, NGS platforms do not output the whole transcripts but a vast set of reads that have to be properly assembled in order to reconstruct transcriptome.
Short-reads assembly represents a crucial point in data analysis, since subsequent steps heavily rely on high quality of reconstructions.
At present there are two main approaches to transcriptome assembly:

- Alignment to reference genome
- Denovo assembly

The first approach is the most efficient and accurate: it consists on alignment of reads to a reference genome, however it is viable only if high quality annotated reference sequences are available for the organism that is object of study.
When references are not available denovo assembly has to be performed: reads are assembled blindly to reconstruct the longer transcripts.
Currently available tools for denovo assembly - such as Bridger [12], Oases [13] and Trinity [14] - emphasise the good sensitivity levels reached, but this result is often obtained by producing an high number of assemblies, consequentially increasing false positives. In addition they have very high hardware requirements, limiting their applicability especially in case of smaller laboratories that do not have access to dedicated computing infrastructures.

False positive reconstructions when working with real data are a very important matter, since in absence of any reference it is not trivial - and maybe not even possibile - to determine the correctness of a reconstruction.

As my MSc thesis project I contributed to the development of a prototype for a new denovo assembler for RNA-seq data, that was specifically designed to overcome current limitations in terms of both false positive reconstructions and hardware requirements. Preliminary results were encouraging so part of my PhD project was aimed towards its finalisation into **STAble** [15] that will be introduced in chapter 4.

1.3 Application of denovo assembly to metagenomics

Metagenomics is the study of uncultured microorganisms sampled directly from their habitat.
NGS based metagenomics approaches allow us to study organisms that live in particular conditions that cannot be reproduced in laboratory environment and to analyse complex interactions that are established among different microbial populations but would otherwise be missed by examining artificial cultures.
This scenario opens further challenges for bioinformaticians, since it is necessary to deal with data coming from heterogenous communities that can be noisy and partial.
A typical metagenomics experiment is composed by two approaches: *microbiota identification* [16] and *metatranscriptome characterisation* [17].
The first is the most exploited one and consists on profiling the microbial population in order to determine which species are actually present in a given sample, alongside their relative abundances.

The latter one is complementary to the former and allows to to understand the effective metabolic activity of the profiled population by analysing collective transcriptome.
Microbiota identification is usually performed by amplicon sequencing 16s rRNA (that is conserved between different species) and to align reads to databases of annotated sequences, such as RDP [18]: in some cases it is possible to perform classification even at species level.
Metatranscriptome characterisation involves a WTSS and at present transcripts reconstruction is usually performed by direct alignment of reads to a reference database, however - given the short length of reads - this approach is expected to cause many assignment ambiguities during mapping process and does not consider the possible present of new unannotated transcripts.

As part of this project we explored the possibility to apply denovo assembly approach to the collective metatranscriptome, by tuning STAble to assemble reads from multiple organisms.
Preliminary results for this extension will be presented in chapter 4 while future perspectives are discussed in chapter 5.

Bibliography

[1] Zhang J, Chiodini R, Badr A, Zhang G. *The impact of next-generation sequencing on genomics.* J Genet Genomics 2011; 38(3): 95–109.

[2] Sanger F, Nicklen S, Coulson AR *DNA sequencing with chain-terminating inhibitors.* Proc Natl Acad Sci USA 1977; 74(12):5463–5467.

[3] L. Feliubadaló, A. Lopez-Doriga, E. Castellsagué et al., *Next-generation sequencing meets genetic diagnostics: Development of a comprehensive workflow for the analysis of BRCA1 and BRCA2 genes.* European Journal of Human Genetics 2013; 21(8):864-70.

[4] Gordon Robertson et al. *De novo assembly and analysis of RNA-seq data.* Nature Methods 7, 2010; 909–912.

[5] Wooley JC, Godzik A, Friedberg I. *A Primer on Metagenomics.* PLoS Comput Biol 2010; 6(2): e1000667.

[6] Anand P, Kunnumakkara AB, Sundaram C, et al. *Cancer is a preventable disease that requires major lifestyle changes.* Pharm Res 2008; 25(9):2097-116

[7] Gage M, Wattendorf D, Henry LR. *Translational advances regarding hereditary breast cancer syndromes.* J Surg Oncol 2012; 105(5):444-51

[8] Szabo C, Masiello A, Ryan JF, Brody LC. *The breast cancer information core: database design, structure, and scope.* Hum Mutat. 2000; 16(2):123-31.

[9] Lontok E, Harrington P, Howe A, Kieffer T, Lennerstrand J, Lenz O, McPhee F, Mo H, Parkin N, Pilot-Matias T, Miller V. *Hepatitis C virus drug resistance-associated substitutions: State of the art summary.* Hepatology. 2015 Nov; 62(5):1623-32.

[10] Qiu P, Stevens R, Wei B, Lahser F, Howe AYM, et al. *HCV Genotyping from NGS Short Reads and Its Application in Genotype Detection from HCV Mixed Infected Plasma.* PLOS ONE 2015; 10(4): e0122082.

[11] In Seok Yang and Sangwoo Kim *Analysis of Whole Transcriptome Sequencing Data: Workflow and Software.* Genomics Inform. 2015 Dec; 13(4): 119–125.

[12] Chang Z, Li G, Liu J, Zhang Y, Ashby C, Liu D, Cramer CL, Huang X. *Bridger: a new framework for de novo transcriptome assembly using RNA-seq data.* Genome Biology 2015; 16:30.

[13] Schulz MH, Zerbino DR, Vingron M, Birney E. *Oases: robust de novo RNA-seq assembly across the dynamic range of expression levels.* Bioinformatics 2012; 28: 1086–1092.

[14] Grabherr MG, Haas BJ, Yassour M, Levin JZ, Thompson DA, Amit I, et al. *Full-length transcriptome assembly from RNA-Seq data without a reference genome.* Nat Biotechnol. 2011; 29: 644–652.

[15] Igor Saggese, Giovanni Manzini and Flavio Mignone. *STAble: a novel approach to denovo assembly of RNA-seq data.* 13th International Meeting CIBB 2016; 2016 Sept 1-3; Stirling - UK.

[16] Shah N, Tang H, Doak TG, Ye Y. *Comparing bacterial communities inferred from 16S rRNA gene sequencing and shotgun metagenomics.*. Methods Mol Biol. 2011; 733:195-205.

[17] Gilbert JA, Hughes M. *Gene expression profiling: metatranscriptomics.* Pac Symp Biocomput 2011; 165-76.

[18] James R. Cole, Qiong Wang, Jordan A. Fish, Benli Chai, Donna M. McGarrell, Yanni Sun, C. Titus Brown, Andrea Porras-Alfaro, Cheryl R. Kuske and James M. Tiedje *Ribosomal Database Project: data and tools for high throughput rRNA analysis.* Nucleic Acids Res. 2014 Jan 1; 42(Database issue): D633–D642.

Chapter 2

Outline of the Thesis

The main aim of this research project was the development of new bioinformatic algorithms and integrated tools to address several aspects of NGS data analysis, with particular focus on three main areas: *molecular diagnostics* [1], *denovo assembly of RNA-seq data* [2] and application of denovo assembly to *metagenomics* [3].

The first part of this thesis is focused on two specific areas of molecular diagnostics, that are the *identification of variants in gene panels* and *virus genotyping*.
In both areas, *variant calling* - that is the process of aligning reads to a reference sequence in order to detect potential variants - represents possibly the most critical step since quality of results is strongly depending on its ability to discriminate between real variants and sequencing errors.
Many variant callers are available for research use, however they are not applicable to clinical routine environments since for diagnostic purposes the entire development process must be carried in compliance with strict requirements for achievement of proper certifications.
Once variants are identified it is also important to annotate them with functional informations and to perform comparison with external resources to get most up-to-date clinical informations about their effects,

however to date this task is not trivial all because of the presence of multiple competing formats for variant nomenclature [4].
Moreover all these steps in clinical environment should be performed within a completely automatised procedure that does not require any manual intervention.
My contribution to this area was the the development of a robust procedure for variant calling and annotation starting from NGS data and a module for automatic integration with external resources that transparently resolves issues caused by different nomenclature conventions. The procedure was extensively validated on both simulated and real clinical data and now constitutes the core of two integrated tools: **Amplicon Suite** and **SmartVir**.
Amplicon Suite is a user-friendly platform for variant calling and annotation on amplicon sequencing NGS data. Thanks to a collaboration with research group of Dr. Ettore Capoluongo of Policlinico Gemelli in Rome we were able to obtain CE-IVD certification for analysis of BRCA1 and BRCA2 genes, that are known to be involved in breast and ovarian cancer.
SmartVir was developed in collaboration with Roche Italian sequencing team and easily allows to perform HCV genotyping and drug resistance detection in order to administer most appropriate treatment to infected patients.
These tools and obtained results will be presented in chapter 3.

Currently available tools for denovo assembly of RNA-seq data - such as Bridger [5], Oases [6] and Trinity [7] - share similar approaches (as they rely on the identification of k-mer sequences) and they achieve high levels of sensitivity at the expense of a consistent number of false positive reconstructions and very high hardware requirements.
As my MSc thesis project I contributed to the development of a new strategy for denovo transcriptome assembly that is based on an original approach where the whole reads are used to drive the assembly process

instead of considering only smaller k-mers with the aim of reducing the number of false positive reconstructions.
Preliminary results were encouraging, so part of my PhD projects was aimed towards its finalisation into **STAble** [8], that will be introduced in chapter 4.

Finally the last part of this thesis presents a contribution to metagenomics, more specifically to the metatranscriptome assembly problem which is very actual while analysis procedures are still at an early stage.
Reconstruction of microbial transcripts from NGS datasets is actually being performed by direct alignment of short reads to reference databases, however this is expected to cause many assignment ambiguities during mapping process and it is possible to miss new unannotated transcripts.
My specific contribution was the tuning of STAble for assembly of transcripts from mixed populations (such as microbial communities), thus bringing the benefits of single organism denovo RNA-seq assembly to metatranscriptome studies. This extension will be presented in chapter 4 while future perspectives are discussed in chapter 5.

Bibliography

[1] L. Feliubadaló, A. Lopez-Doriga, E. Castellsagué et al., *Next-generation sequencing meets genetic diagnostics: Development of a comprehensive workflow for the analysis of BRCA1 and BRCA2 genes.* European Journal of Human Genetics 2013; 21(8):864-70.

[2] Gordon Robertson et al. *De novo assembly and analysis of RNA-seq data.* Nature Methods 7, 2010; 909–912.

[3] Wooley JC, Godzik A, Friedberg I. *A Primer on Metagenomics.* PLoS Comput Biol 2010; 6(2): e1000667.

[4] Shuji Ogino, Margaret L. Gulley, Johan T. den Dunnen, Robert B. Wilson and the Association for Molecular Pathology Training and Education Committee. *Standard Mutation Nomenclature in Molecular Diagnostics.* J Mol Diagn. 2007 Feb; 9(1): 1–6.

[5] Chang Z, Li G, Liu J, Zhang Y, Ashby C, Liu D, Cramer CL, Huang X. *Bridger: a new framework for de novo transcriptome assembly using RNA-seq data.* Genome Biology 2015; 16:30.

[6] Schulz MH, Zerbino DR, Vingron M, Birney E. *Oases: robust de novo RNA-seq assembly across the dynamic range of expression levels.* Bioinformatics 2012; 28: 1086–1092.

[7] Grabherr MG, Haas BJ, Yassour M, Levin JZ, Thompson DA, Amit I, et al. *Full-length transcriptome assembly from RNA-*

Seq data without a reference genome. Nat Biotechnol. 2011; 29: 644–652.

[8] Igor Saggese, Giovanni Manzini and Flavio Mignone. *STAble: a novel approach to denovo assembly of RNA-seq data.* 13th International Meeting CIBB 2016; 2016 Sept 1-3; Stirling - UK.

Chapter 3

Analysis workflows for NGS

The advent of NGS techniques paved the road for new research areas and revolutionised many existing ones. Recent advancements allow us to sequence nucleic acids at increasingly high speeds and accuracy while lowering the cost per base.

Molecular diagnostics is a very actual area to which NGS has opened new frontiers: this name identifies a collection of techniques aimed at identifying specific markers in a patient's genome and transcriptome and to analyse their effects on expressed proteins.

Current applications of NGS to this area are offering the perspective of personalised medicine, with the possibility of modelling specific therapies for each person. However there are still open challenges - especially in the bioinformatics analysis side - that are delaying the integration of these techniques into routine clinical procedures and they are being applied only as a support to traditional ones.

Our contribution to molecular diagnostics is focused on two specific areas: *identification of variants in gene panels* and *virus genotyping*.

A strong *variant calling* mechanism is the core of most bioinformatics analysis procedures for molecular diagnostics: it allows the identification of nucleotide variants by aligning NGS reads to a reference sequence as shown in Figure 3.1.

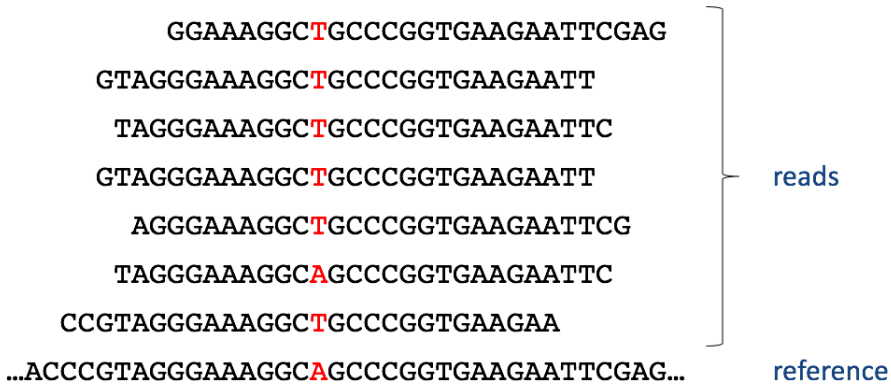

Figure 3.1: Representation of variant calling process where short NGS reads are aligned to a reference sequence. In this example 6 out of 7 reads have a T instead of an A at the highlighted position, so there is a potential variant.

The main problem that must be taken into account in this process it the presence of sequencing errors: the accuracy of a variant calling mechanisms resides in its ability to discriminate between them and real variants by applying proper statistics.
The typical subsequent step is *variant annotation* that consists on assignment of functional informations to detected variants: depending on their location they are likely to have a different impact on cell's health. Variants in intronic regions might cause no visible effects, but when they alter splicing sites or coding sequences it is desirable to study them more in depth.
Finally, detected and annotated variants should be compared to external resources, like databases and functional predictors, to get additional informations about their effects (such as if they are known to be linked to a particular pathology). Even though this task may look straight-forward at first glance, at present it is not trivial at all, mainly because of the lack of commonly agreed standards for variants nomenclature [1]. *Human Genome Variation Society (HGVS)* back in the year 2000 proposed its own recommendations [2]: despite they

have been widely adopted and are likely to become the international standard, to date there are still many resources that implement alternative formats.
Given these premises, a new workflow that implements all these steps was developed and used as core for two integrated tools: **Amplicon Suite** and **SmartVir**.
Amplicon Suite is a platform for analysis of molecular diagnostics datasets that can be easily extended to work with all annotated genes.
SmartVir was developed in collaboration with Roche Italian sequencing team and performs fast and accurate HCV genotyping and drug resistance detection.
Both tools can be used through an intuitive graphical user interface (GUI) carefully designed to allow users with no specific bioinformatics skills to perform analyses autonomously.
The new pipeline will be introduced and then its specific implementation into the two integrated tools will be discussed.

3.1 A new pipeline for variant calling and annotation

The new pipeline for variant calling and annotation is depicted in Figure 3.2: it expects raw NGS amplicon sequencing data as input and returns a set of detected and annotated variants.

3.1.1 Analysis profiles

The new pipeline is designed to be flexible and to work with NGS amplicon sequencing data from any annotated gene.
We defined an automated procedure for the creation of a proper *profile* that can be used to introduce support for new genes. A profile is composed by reference sequences to be used for alignment (*NCBI Ref-*

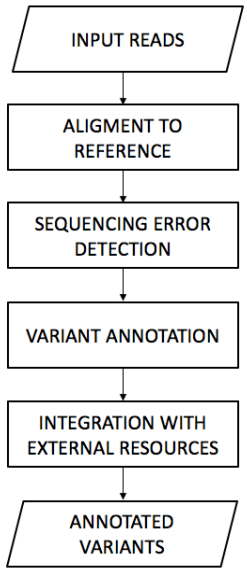

Figure 3.2: The new variant calling and annotation pipeline.

Seq data [3]) and other informations required to complete all analysis steps. Most relevant informations involved are:

- List of genes on which amplicons where designed.
- Primer sequences used for amplification.
- List of transcripts related to target genes and their annotation.
- Informations about amplification and sequencing kit.

3.1.2 Alignment to reference

This module accepts a NGS amplicon sequencing dataset as input that is analysed in order to detect potential variants.
The procedure is depicted in Figure 3.3 and consists of two steps:

1. Detection of appropriate reference sequence and its sub-region.
2. Accurate alignment to sub-region for variant detection.

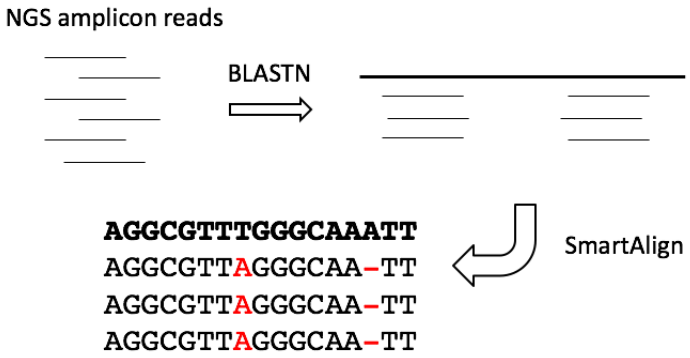

Figure 3.3: The two-step alignment procedure. Short reads are mapped to correct reference sequence using BLASTN and then are more accurately aligned to a specific region using SmartAlign.

In the first step reads are mapped to genes listed in analysis profile using BLASTN [4], similarity cutoff to adopt is parametric and should be carefully chosen depending on the specific application.
Afterwards each read is more accurately aligned to its mapping region using **SmartAlign**, a custom aligner developed by my research group that implements the *Needleman-Wunsch* dynamic programming algorithm [5], with penalty scores carefully tuned for NGS data.
We chose to develop our custom aligner to get full control of the situation since, as shown in Figure 3.4, this step is critical: there might be multiple algorithmically correct alignments, however their biological meaning can be significantly different.

The final output of this step is the set of potential variants detected for each read, expressed in genomic absolute coordinates with the following notation:

$$\langle \text{chr\_name} \rangle : \langle \text{chr\_pos} \rangle : \langle \text{nt\_ref} \rangle : \langle \text{nt\_var} \rangle$$

- **chr_name:** name of the chromosome where the gene is located.
- **chr_pos:** genomic start position.

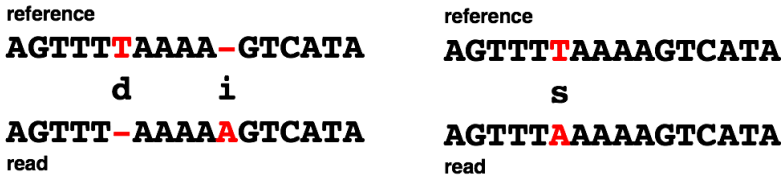

Figure 3.4: Both alignments shown in the example are algorithmically correct but their biological meaning is very different. Alignment on the left implies an insertion and a deletion in the same read, which is an unlikely event. The one on the right involves a single substitution, which seems more reasonable from a biological perspective. However if we suppose that read is generated by Roche 454 platform, which has known issues with homopolymers, we can state with a good degree of confidence that alignment on the left is the correct one, but both variants are not real and have to be marked as sequencing errors.

- **nt_ref:** original allele from reference
- **nt_var:** allele found in read.

This notation contains all required informations to unambiguously identify each variant, however it is just for internal use: there are multiple plugins that before generating final output convert it to user's preferred format, with particular emphasis on HGVS standards. Here are some examples of real variant names:

chr13:32930761:A:G (substitution of A with G)
chr13:32900613:-:T (insertion of T)
chr17:41251858:T:- (deletion of T)
chr13:32906457:ACCACAT:------- (deletion of ACCACAT)

3.1.3 Sequencing error detection

This module filters variants reported by previous one by applying proper statistics with the aim of removing potential sequencing errors.

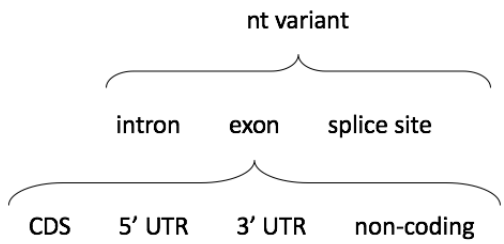

Figure 3.5: Hierarchy used by variant annotation module for classification.

Main parameters involved in filtering are the number of supporting reads and mean quality score of involved bases. Also a minimum coverage for target region is required.
Actual values for these thresholds must be chosen according to requirements of specific application.

3.1.4 Variant annotation

Variant annotation module classifies nucleotide variants according to their effect on expressed transcripts: classification hierarchy is depicted in Figure 3.5.
There is a first grouping based on the involved region, that can be an *intron*, an *exon* or a *splice site* in the boundary between two of them. In case of intronic regions, the variant is just labeled that way and no further actions are taken, since it is not likely to produce any effect.
When the variant modifies a splice site between an intron and an exon, a warning is raised and both the original and modified versions of the site will be reported as the situation needs to be carefully evaluated.
Finally, in case of exonic regions there is an additional layer of classification:

- **CDS:** variant modifies an annotated coding region.
- **5' UTR:** variant modifies 5' UTR region.

- **3' UTR:** variant modifies 3' UTR region.
- **non-coding:** affected transcript is non-coding.

Nucleotide variants that affect a CDS are also annotated with their representation at amino-acid level:

$$\text{p.}\langle\text{aa\_ref}\rangle\langle\text{aa\_pos}\rangle\langle\text{aa\_var}\rangle$$

- **aa_ref:** the original amino acid from the reference sequence.
- **aa_pos:** amino acid position in the protein sequence.
- **aa_var:** the variated amino acid.

Some examples of real variants:

p.K21S (substitution of K with S in position 21)
p.L35del (deletion of L in position 35)
p.15-16insV (insertion of V after position 15)

3.1.5 Integration with external resources

Interoperability with external dedicated resources is nowadays a very important feature, since it allows to get access to the latest discoveries from the scientific community.
Many widely used authoritative resources are available for molecular diagnostics, from functional predictors for mutations to dedicated databases where knowledge about the effect of known variants resides. Unfortunately accessing them can be as important as tricky, mainly because universal standards for variant nomenclature are still far from being fully deployed. In addition, some databases do not explicitly state accession number and revision of the reference sequence used for annotation, so there is a considerable risk of performing the wrong comparisons.

To address this issue, the new pipeline is not designed around a particular nomenclature but internally uses its own format for variant representation and implements multiple plugins that are able to perform conversions to different formats when outputting data to the user, with particular attention to HGVS recommendations that are used as default.
The same plugins are also used to perform automatic integration with external resources, that can be queried by performing a name conversion from our internal format to the accepted one.

3.2 Amplicon Suite

Amplicon Suite is a user-friendly tool for variant calling and annotation from NGS amplicon sequencing data. It is designed to allow clinicians and researches to perform analyses autonomously without requiring specific bioinformatics skills.
It is based on the thin-client approach where the user can interact with the system through an intuitive graphical user interface, while the analysis core resides on a remote server. This architecture was chosen to deal with typical restrictions on hospital networks where personal computers may not be not powerful enough to run analysis process and strict policies about installed software are enforced. In addition, to address legal privacy issues it is also possible to install analysis core on a dedicated server connected to the local network, thus not requiring upload of data to remote hosts.
By setting up a proper profile it is possible to work with any annotated gene, however thanks to a collaboration with Policlinico Gemelli in Rome we had the opportunity to focus on the setup and tuning of a profile specific for analysis of variants on BRCA1 and BRCA2 genes, that are known to be involved in breast and ovarian cancer.
The analysis workflow is shown in Figure 3.6 and is based on the new

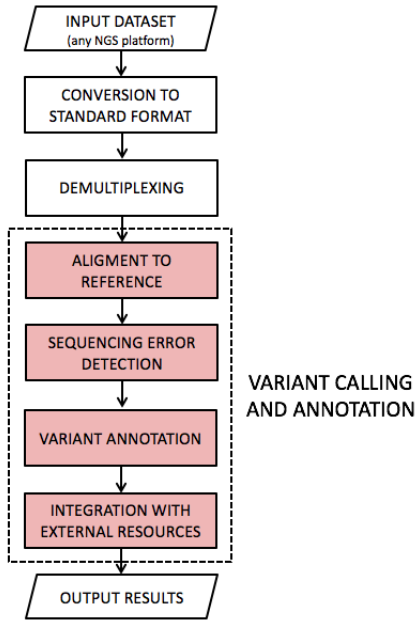

Figure 3.6: Amplicon Suite's analysis workflow. Modules that implement the new variant calling pipeline are highlighted in red.

variant calling pipeline.

Data input The user uploads a raw NGS dataset through an intuitive graphical user interface and is asked to answer few simple questions about the procedure used for sample preparation. Currently accepted input formats are *SFF* files returned by Roche 454 and *FASTQ* files from Illumina and IonTorrent platforms.

Conversion to standard format Amplicon Suite is able to analyse datasets returned from the major sequencing platforms available and adding support for new ones is as simple as extending this module.
Input is converted to a standardised format, that consists of a *FASTA* file with raw sequences and a *QUAL* file where quality scores annotated with *Phred* values [6].
A metadata file is also generated, containing informations required

by subsequent steps, such as used sequencing platform and its error model.

Demultiplexing Current NGS platform have a very high throughput that in most cases exceeds actual needs. Because of this, a common approach is to perform *multiplexing* and sequence multiple samples in a single run.
Specific multiplexing protocol employed varies depending on the sequencing platform that will be used, however it is usually achieved by incorporating specific *barcodes* (that in Roche 454 environment are usually called *MIDs*) at the ends of amplified sequences. Every sample is associated to a specific barcode so during analysis process it is possible to perform demultiplexing by detecting them.
In case of Illumina or IonTorrent datasets, sequencing output is already demultiplexed and a separate FASTQ file for each sample is provided.
Conversely, for Roche 454 datasets MIDs at the ends of each read are detected and separate pairs of FASTA and QUAL files are generated for each sample.

Variant calling and annotation This macro-step consists of the implementation of the new pipeline. Demultiplexed reads are at first mapped to BRCA1 and BRCA2 references (NG_005905.2 and NG_012772.1) using BLASTN in order to perform assignment of each input sequence to the correct one.
SmartAlign is then applied to perform a more accurate alignment of each read to the mapping region to detect potential variants that are subsequently filtered for sequencing error removal and annotated over expressed transcripts.
Finally, integration module is currently implemented for 4 resources:

- **Breast Cancer Information Core (BIC):** an open access

online breast cancer mutation database [7].

- **dbSNP:** the NCBI database of genetic variation [8].
- **SIFT:** predictor for the effects of amino acid substitution on protein function [9].
- **Polyphen:** predictor of functional effects of human nsSNPs [10].

Output results Results are reported to the user through the graphical user interface, in form of tables with detected variants and charts related to their frequencies and amplicon coverage.
Clinicians can review results and select which mutations are to appear in the final report, that can be saved in PDF format and printed.

3.2.1 Clinical validation

Amplicon Suite was clinically validated and CE-IVD certified for the analysis of BRCA1 and BRCA2 genes in collaboration with research group of Dr. Ettore Capoluongo of Policlinico Gemelli in Rome.
Validation was performed by integration into the analysis workflow proposed in Figure 3.7, that describes a complete protocol (from sample preparation to NGS data analysis) for routine analysis of BRCA full exome and exon-intron flanking regions.
The whole procedure is based on the use of a multiplex PCR strategy [11] (BRCA MASTR kit by Multiplicom [12]) that is able to generate DNA library, followed by 454 GS Junior pyrosequencing [13].
Due to technical limitations of pyrosequencing in deciphering homopolymer stretches [14], a specific validated pre-NGS quality control step based on fragment analysis [15] that is able to evaluate quality of PCR multiplex and identify small indels in coding regions has been setup.
The whole procedure is documented in [16], however here we'll focus on bioinformatics analysis step.

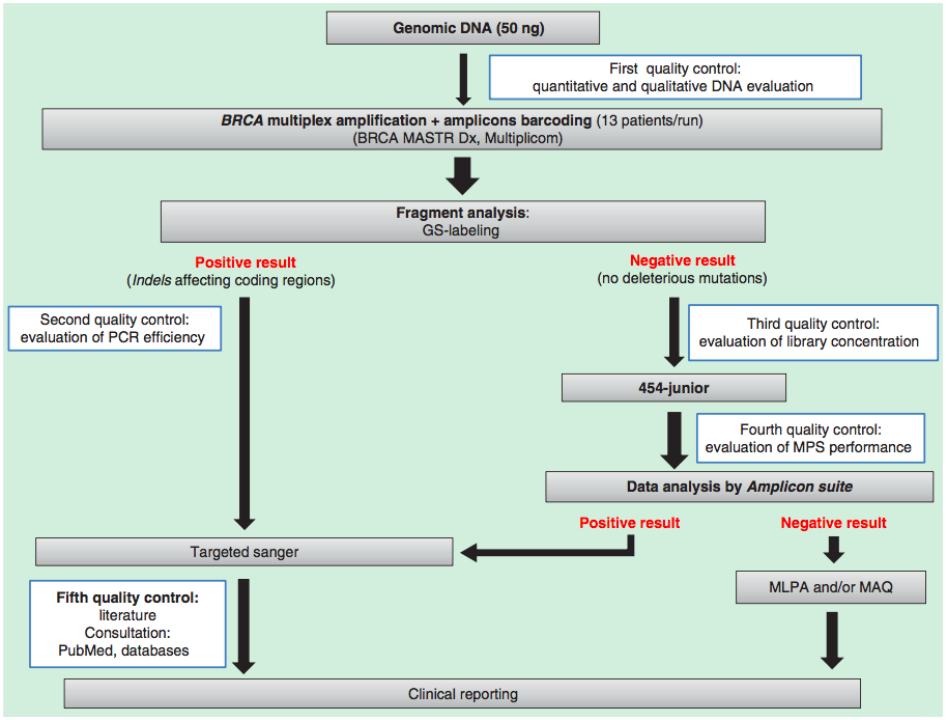

Figure 3.7: Integrated NGS workflow used for validation of Amplicon Suite (image from [16]).

Validation method

Validation was performed on 220 samples from women that were diagnosed as sporadic and/or familial ovarian cancer patient.
Genomic DNA was isolated from peripheral blood by a method based on a commercial kit distributed by Roche Diagnostics.
Samples were then PCR-enriched using Multiplicom BRCA MASTR Dx assay v2.0, that covers all the coding regions and splice sites of BRCA genes with 93 amplicons per patient.
Sequencing was performed on GS Junior platform (23 runs in total) with GS Junior Titanium Sequencing Kit in combination with the matching GS Junior Titanium PicoTiterPlate (PTP).
The first 10 runs (80 samples in total) were analysed with both *Amplicon Variant Analyzer (AVA)* and Amplicon Suite, in order to allow validation of the new software, while the remaining 13 were analysed primarily with Amplicon Suite and only subsequently with AVA.
AVA is part of the software suite distributed by Roche Diagnostics with every GS sequencing platform and is specifically designed for analysis of amplicon sequencing experiments, however it has some important limitations. Noteworthy, advanced coverage analysis is of primary importance in diagnostic setting, but it cannot be achieved by using AVA and data is not stored in a structured way (e.g. flat files instead of a database) not allowing to easily compare results of different runs and to create for instance an historic database of all variants of interest that were detected. Moreover the process of creation of a new analysis project can be tricky for a user with no specific bioinformatic skills and automatic integration of external resources is not provided: Amplicon Suite was specifically designed around user requests to overcome all these issues.
Reference sequences NG_005905.2 (BRCA1) and NG_012772.1 (BRCA2) were used for alignment and results were confirmed by sequencing all samples also with Sanger, that is actually considered as the golden

standard for these applications.
Analysis results have been compared against most relevant databases for clinical use: dbSNP [8], BIC [7], LOVD [1], UMD [2], AURP [3] and HGMD [17].

Evaluation of amplicon coverage

The first task was the evaluation of amplicon coverage for each patient. Two threshold values were established: 30x and 38x that were arbitrarily defined as *alert* and *enhanced* level.
In literature 30x is actually considered as the minimum acceptable coverage for genomic clinical studies [18], while in the setup pipeline 38x indicates an optimised performance level.
Table 3.1 reports obtained results by Amplicon Suite for *whole amplicon coverage* that indicates the number of patients for which at least the threshold values of 30x or 38x are overcome in all 93 amplicons.
This information was extremely useful to tune analysis protocol: data shows that the performance of amplicons not reaching fixed coverages did not depend on the number of samples loaded in each run, since with 13 samples per run the number of amplicons that failed the target values decreased.

Indel identification by fragment analysis

Given the known limitations of pyrosequencing in deciphering homopolymer stretches, indel identification was performed separately by fragment analysis based on *Multiplex ligation-dependent probe amplification (MLPA)* [19], which is a variation of the multiplex polymerase chain reaction that permits amplification of multiple targets with only a single primer pair and it is widely used to determine relative ploidy.

[1] http://www.lovd.nl/3.0/home/
[2] http://www.umd.be/
[3] http://www.arup.utah.edu/database/

Table 3.1: *Whole amplicon coverage* values reported by Amplicon Suite (data from [16]).

Subjects per run	# of runs	Subjects (%)	Alert level $> 30x$	Enhanced level $> 38x$
8	15	120 (54)	87 (73)	74 (62)
11	1	11 (5)	5 (46)	6 (55)
12	2	24 (11)	13 (54)	12 (50)
13	5	65 (30)	53 (82)	28 (43)
Total	23	220 (100)	158 (72)	120 (55)

Probes are designed to target genes of interest and their signal strengths are compared with those obtained from a reference DNA known to have two copies of the chromosome. If an extra copy is present in the test sample, the signals are expected to be 1.5 times the intensities of the respective probes from the reference. If only one copy is present the proportion is expected to be 0.5. If the sample has two copies, the relative probe strengths are expected to be equal.
Results from Table 3.2 show that 31 out of 212 samples resulted positive in this test and also gives an insight about the complexity of implementing automatic integration with external resources: in this list, nomenclature for the same variant is very different between HGVS recommendations and BIC annotation.
Since the presence of two deleterious mutations is a very rare event in BRCA genes, these samples were not sequenced by NGS, but immediately sequenced by targeted Sanger: perfect concordance between fragment analysis and Sanger for all the prescreened indels was obtained.
In addition the accuracy of Amplicon Suite has been confirmed since it reported no further indels for those samples that resulted negative to this test and were processed with NGS.

Table 3.2: List of 21 indels found in 31 patients by fragment analysis before NGS run (data from [16]).

# of carriers	Gene	Exon	HGVS nucleotide	BIC nucleotide	HGVS protein	dbSNP	Clinical importance					
							BIC	LOVD	UMD	AURP	HGMD	Extra
1	BRCA1	2	c.66_67delAG	185delAG	p.L22_E23LVfs	80357713	Yes	-	-	Yes	Breast cancer	-
1	BRCA1	11	c.843_846delCTCA	962delCTCA	p.S281_S282?fs	80357919	Yes	-	Yes	Yes	Breast cancer	-
1	BRCA1	11	c.850_851insTCATTAC	969insTCATTAC	p.Q284?fs	80357989	Yes	-	-	-	Breast cancer	-
1	BRCA1	11	c.1898_1899insTT AAGCCCACAAAT	2017insTT AAGCCCACAAAT	p.P634Qfs	-	-	-	-	-	-	Novel
1	BRCA1	11	c.1961_1961delA	2080delA	p.K654Sfs	80357522	Yes	-	Yes	Yes	Breast cancer	-
1	BRCA1	11	c.2269_2269delG	2388delG	p.V757F?fs	80327583	Yes	-	Yes	Yes	Breast cancer	-
1	BRCA1	11	c.3173_3176delTAAT	3192delTAAT	p.I1058Mfs	-	-	-	-	-	-	Novel
3	BRCA1	11	c.3756_3759delGTCT	3875delGTCT	p.S1253Rfs	80357868	Yes	-	Yes	Yes	Breast cancer	-
1	BRCA1	12	c.4165_4166delAG	4284delAG	p.S1389X	80357572	Yes	-	Yes	Yes	Breast cancer	-
1	BRCA1	17	c.5062_5064delGTT	5181delGTT	p.V1688del	80358344	Yes	Yes	Yes	Yes	Breast and/or ovarian cancer	-
7	BRCA1	20	c.5263_5264insC	5382insC	p.S1755?fs	80357906	Yes	Yes	Yes	-	-	-
2	BRCA2	10	c.1238_1238delT	1466delT	p.L413Hfs	80359271	Yes	-	-	Yes	Breast and/or ovarian cancer	-
1	BRCA2	10	c.1796_1800delTTTAT	2024delTTTAT	p.S599_Y600?fs	-	Yes	-	-	-	-	-
1	BRCA2	11	c.3192_3195delAATT	3420delAATT	p.Ser1064_Ile1065?fs	80359375	-	-	-	-	-	-
1	BRCA2	11	c.3683_3684insG	3911insG	p.Asn1228K?fs	-	-	-	-	-	-	Novel
2	BRCA2	11	c.3744_3747delTGAG	3972delTGAG	p.S1248Rfs	80359403	Yes	-	Yes	Yes	Breast and/or ovarian cancer	-
1	BRCA2	11	c.4282_4283insT	4510insT	p.F1428?fs	80359439	Yes	-	-	Yes	-	-
1	BRCA2	11	c.6313_6313delA	6541delA	p.I2105?fs	80359439	-	-	-	-	-	Novel
1	BRCA2	15	c.7498_7498delA	7726delA	p.R2500?fs	-	-	-	-	-	-	Novel
1	BRCA2	19	c.8463_8464insT	8691insT	p.I2822Tfs	-	-	-	Yes	-	-	-
1	BRCA2	25	c.9413_9414insT	9641insT	p.L3138Ffs	-	-	-	-	Yes	-	-

Table 3.3: BRCA deleterious mutations identified by NGS and confirmed by targeted Sanger sequencing (data from [16]).

# of carriers	Gene	Exon	HGVS nucleotide	BIC nucleotide	HGVS protein	dbSNP	Clinical importance				
							BIC	LOVD	UMD	AURP	HGMD
2	BRCA1	5	$c.181T > G$	$300T > G$	p.C61G	28897672	Yes	-	Yes	Yes	Breast cancer
1	BRCA1	11	$c.3257T > G$	$3376T > G$	p.L1086X	80357006	Yes	-	Yes	Yes	Breast cancer
1	BRCA1	11	$c.3514G > T$	$3633G > T$	p.E1172X	-	-	-	-	Yes	Breast and/or ovarian cancer
1	BRCA1	12	$c.4117G > T$	$4236G > T$	p.E1373X	80357259	Yes	Yes	Yes	Yes	Ovarian cancer
1	BRCA1	13	$c.4258C > T$	$4377C > T$	p.Q1420X	80357305	Yes	-	Yes	Yes	Breast cancer
1	BRCA1	18	$c.5123C > A$	$5242C > A$	p.A1708E	28897696	Yes	Yes	Yes	Yes	Breast cancer
1	BRCA1	19	$c.5161C > T$	$5280C > T$	p.Q1721X	-	-	-	Yes	Yes	-
3	BRCA2	IVS10	$c.1909 + 1G > A$	$2137 + 1G > A$	-	-	-	-	Yes	-	-
1	BRCA2	16	$c.7681C > T$	$7909G > T$	p.Q2561X	80358994	Yes	-	-	-	-
1	BRCA2	17	$c.7857G > A$	$8085G > A$	p.W2619X	80359011	Yes	-	-	Yes	Breast and/or ovarian cancer

Table 3.4: VUS identified by NGS and confirmed by targeted Sanger sequencing (data from [16]).

# of carriers	Gene	Exon	HGVS nucleotide	BIC nucleotide	HGVS protein
1	BRCA1	11	$c.804C > G$	$923C > G$	p.N268K
1	BRCA1	11	$c.2501G > A$	$2620G > A$	p.G834E
1	BRCA1	11	$c.3868A > G$	$3987A > G$	p.K1290E
1	BRCA1	14	$c.4361T > C$	$4480T > C$	p.V1454A
1	BRCA1	17	$c.5058T > A$	$5177T > A$	p.H1686Q
1	BRCA2	10	$c.1444C > T$	$1672C > T$	p.L482F
1	BRCA2	25	$c.9383C > T$	$9601C > T$	p.L3125F

Detection of mutations with NGS

All samples that resulted negative to the fragment test where sequenced with GS Junior platform and results compared against reference clinical databases (integration with BIC and dbSNP was automatically performed by Amplicon Suite, while for the other resources results were manually compared).
Detected variants and mutations are are subdivided into 3 main groups according to their clinical relevance:

- **Deleterious mutations:** mutations that are known to produce harmful effects.
- **Variants of uncertain significance (VUS):** novel mutations or variants for which informations included in different databases were discordant regarding the pathogenic effect at both biological and clinical level.
- **Variants of no significance:** synonymous SNPs or missenses that all interrogated databases concordantly classified as not producing noteworthy effects.

Among 181 samples analysed by NGS, 10 different deleterious mutations (see Table 3.3) were found in 13, including 7 BRCA1 mutations

(5 nonsense and 2 missense) and 3 BRCA2 mutations (2 nonsense and 1 missense).
VUS were detected in 7 patients and are reported In Table 3.4. One patient carried the p.K1290E along with the deleterious BRCA2 exon 20 deletion, while the remaining ones were identified in women not carrying other deleterious mutations.
Finally, 55 synonymous and missense SNPs were found: 25 and 32 in BRCA1 and BRCA2 genes, respectively. The exhaustive list will not be reported here but is available in [16].
Most importantly, all mutations reported by Amplicon Suite were confirmed by targeted Sanger sequencing, demonstrating that it is possible to apply it for routine clinical analysis of BRCA mutations.

3.3 SmartVir

NGS techniques allow mass sequencing of viral genome, providing the opportunity to probe viral population from a single host.
There is a growing interest in these applications for both research and clinical purposes, however dedicated resources - such as databases with reference genotypes and drug resistance related mutations - are still fragmented, making the creation of an integrated solution for routine analysis not trivial at all.
As our contribution to this field we developed **SmartVir**, a user-friendly integrated tool for virus genotyping and drug resistance detection. Like Amplicon Suite it is designed to be generic and to work with many viral species, however within a collaboration with Roche Italian Sequencing team we focused on the creation of a profile for the treatment of HCV infections.
Analysis workflow is depicted in Figure 3.8: it shares some modules with Amplicon Suite (input dataset, conversion to standard format and demultiplexing) and it is also based on the new variant calling

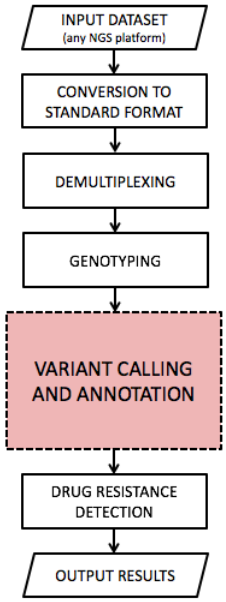

Figure 3.8: SmartVir's analysis workflow.

pipeline.

Genotyping This module performs genotyping of each sample in order to detect which strain (or strains in case of mixed infections) is affecting the patient.
Applied technique is the one introduced in [20] that allows discrimination between the first six genotypes (1, 1a, 1b, 2, 3, 4, 5 and 6) without ambiguities.
Reads from each sample are aligned using BLASTN to references of all genotypes but only alignments that span over a specific 250-bp area in NS3 (position 701-950), NS5A (position 1-250) and NS5B (position 101-350) regions are retained. Assigned genotype is the one that shows the best matches against all these three sub-regions.

Variant calling and annotation Reads for each sample are subjected to the two-step alignment process: they are first mapped to the

correct region of the assigned reference using BLASTN and then are accurately aligned with SmartAlign for variant detection.
Detection of mutations in viral genome represents an important step towards the choice of the most appropriate treatment.

Drug resistance detection This module implements automatic integration with external resources in order to verify if mutations detected in viral genome are known to cause resistance to certain drugs. Databases that are currently used as reference are *Geno2Pheno* [21] (a web-based decision support system for HCV treatment) and the list of clinically relevant mutations reviewed in [22] that we informally named *Lontok*.

Output results Detected mutations and drug resistances are available in form of tables through the user interface alongside other informations about the quality of the sequencing run (e.g. amplicon coverage and number of reads for sample). It is also possible to export and print a report with a summary of relevant mutations found for each patient.

3.3.1 Results

SmartVir was validated in collaboration with Roche Italian sequencing team starting from a set of 40 HCV samples subdivided in 4 GS Junior runs (10 samples per run).
Sequencing kit is developed by Roche and targets NS3, NS5A and NS5B regions for genotypes 1a, 1b and 3a (which are the most prevalent strains worldwide [23]) over 27 total amplicons.
NS3 is a multifunctional protein with both serine protease and RNA helicase/NTPase activities, NS5A is a phosphoprotein which takes part in virus particle formation and is involved in virus resistance against interferons [24] and NS5B protein encodes for an RNA-dependent

Table 3.5: Demultiplexing and genotyping report for HCV test run.

Sample ID	Genotype	Reference	Total reads	Aligned reads
1	1b	D90208.1	421	376
2	1b	D90208.1	2795	2764
3	1b	D90208.1	5264	4260
4	1b	D90208.1	13392	11450
5	1b	D90208.1	3904	2889
6	1a	AF009606.1	6884	6796
7	1a	AF009606.1	9939	9840
8	1a	AF009606.1	17324	16723
9	1a	AF009606.1	4464	4408
10	1a	AF009606.1	10983	10887

RNA polymerase (RdRp), which is the central catalytic enzyme of the HCV replicase [25, 26].

Results for one of validation runs will be presented.

Table 3.5 reports genotype and reference sequence automatically assigned by SmartVir for each sample alongside alignment statistics (in terms of total reads and reads successfully aligned to reference). Sample 1 has a very low read count and SmartVir immediately raised a warning about this, informing the user that not enough data is available to produce reliable results (it was later confirmed that sample was sent to sequencing despite some amplification problems have occurred).

After quality control check and genotype assignment, SmartVir proceeds to detect mutations on viral genome and automatically compare them to external resources (as already mentioned Geno2Pheno [21] and Lontok [22] are currently integrated) in order to determine possible drug resistances. Samples 1, 9 and 10 resulted in a negative outcome, while drug resistance associated mutations where detected for all the others over NS3 regions (see Table 3.6).

Table 3.6: Drug resistance associated mutations detected by SmartVir.

Sample ID	Region	Mutation	Read count	Total coverage	Frequency	Geno2Pheno	Lontok
2	NS3	S122N	22	1187	1.9%	-	Asunaprevir
3	NS3	S122T	53	3360	1.6%	-	Asunaprevir Simeprevir
4	NS3	D168E	376	6850	5.5%	Asunaprevir Grazoprevir Paritaprevir Simeprevir	Asunaprevir Simeprevir Vaniprevir
5	NS3	S122N	6	2959	0.2%	-	Asunaprevir
5	NS3	R117H	6	2959	0.2%	Telaprevir	-
6	NS3	F43L	24	3971	0.6%	Paritaprevir	-
6	NS3	S122R	23	2912	0.8%	Simeprevir	Simeprevir
6	NS3	R117H	6	2912	0.2%	Telaprevir	-
7	NS3	F43L	4173	4296	97.1%	Paritaprevir	-
7	NS3	S122R	75	5640	1.3%	Simeprevir	Simeprevir
8	NS3	Q80H	4352	4441	98.0%	Simeprevir	-

Bibliography

[1] Shuji Ogino, Margaret L. Gulley, Johan T. den Dunnen, Robert B. Wilson and the Association for Molecular Pathology Training and Education Committee. *Standard Mutation Nomenclature in Molecular Diagnostics.* J Mol Diagn. 2007 Feb; 9(1): 1–6.

[2] den Dunnen JT, Dalgleish R, Maglott DR, Hart RK, Greenblatt MS, McGowan-Jordan J, Roux AF, Smith T, Antonarakis SE, Taschner PE. *HGVS Recommendations for the Description of Sequence Variants: 2016 Update.* Hum Mutat. 2016 Jun; 37(6): 564-9.

[3] O'Leary NA, Wright MW, Brister JR, Ciufo S, Haddad D, McVeigh R, Rajput B, Robbertse B, Smith-White B, Ako-Adjei D, Astashyn A, Badretdin A, Bao Y, Blinkova O, Brover V, Chetvernin V, Choi J, Cox E, Ermolaeva O, Farrell CM, Goldfarb T, Gupta T, Haft D, Hatcher E, Hlavina W, Joardar VS, Kodali VK, Li W, Maglott D, Masterson P, McGarvey KM, Murphy MR, O'Neill K, Pujar S, Rangwala SH, Rausch D, Riddick LD, Schoch C, Shkeda A, Storz SS, Sun H, Thibaud-Nissen F, Tolstoy I, Tully RE, Vatsan AR, Wallin C, Webb D, Wu W, Landrum MJ, Kimchi A, Tatusova T, DiCuccio M, Kitts P, Murphy TD, Pruitt KD. *Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation.* Nucleic Acids Res. 2016 Jan 4; 44(D1):D733-45.

[4] Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. *Basic local alignment search tool.* J Mol Biol. 1990 Oct 5; 215(3):403-10.

[5] Saul B. Needleman, Christian D. Wunsch. *A general method applicable to the search for similarities in the amino acid sequence of two proteins.* Journal of Molecular Biology 1970; 48 (3): 443–53.

[6] Ewing B, Hillier L, Wendl MC, Green P. *Base-calling of automated sequencer traces using phred. I. Accuracy assessment.* Genome research 1998; 8 (3): 175–185.

[7] Szabo C, Masiello A, Ryan JF, Brody LC. *The breast cancer information core: database design, structure, and scope.* Hum Mutat 2000; 16(2):123-31.

[8] S. T. Sherry, M.-H. Ward, M. Kholodov, J. Baker, L. Phan, E. M. Smigielski and K. Sirotkin. *dbSNP: the NCBI database of genetic variation.* Nucleic Acids Res. 2001 Jan 1; 29(1): 308–311.

[9] Ng PC, Henikoff S. *Predicting deleterious amino acid substitutions.* Genome Res. 2001 May; 11(5):863-74.

[10] Ivan Adzhubei, Daniel M. Jordan, and Shamil R. Sunyaev. *Predicting Functional Effect of Human Missense Mutations Using PolyPhen-2.* Curr Protoc Hum Genet. 2013 Jan; 0 7: Unit7.20.

[11] Markoulatos P, Siafakas N, Moncany M. *Multiplex polymerase chain reaction: a practical approach.* J Clin Lab Anal. 2002;16(1):47-51.

[12] Badoer C, Garrec C, Goossens D, Ellison G, Mills J, Dzial M, El Housni H, Berwouts S, Concolino P, Guibert-Le Guevellou V, Delnatte C, Del Favero J, Capoluongo E, Bézieau S. *Performance of multiplicom's BRCA MASTR Dx kit on the detection of BRCA1*

and BRCA2 mutations in fresh frozen ovarian and breast tumor samples. Oncotarget. 2016 Oct 25.

[13] Harrington CT, Lin EI, Olson MT, Eshleman JR. *Fundamentals of pyrosequencing.* Arch Pathol Lab Med. 2013 Sep;137(9):1296-303.

[14] Susan M Huse, Julie A Huber, Hilary G Morrison, Mitchell L Sogin and David Mark Welch. *Accuracy and quality of massively parallel DNA pyrosequencing.* Genome Biol. 2007; 8(7): R143.

[15] Concolino P, Costella A, Minucci A, et al. *A preliminary Quality Control (QC) for next generation sequencing (NGS) library evaluation turns out to be a very useful tool for a rapid detection of BRCA1/2 deleterious mutations.* Clin Chim Acta 2014; 437:72-7

[16] Minucci A, Scambia G, Santonocito C, Concolino P, Canu G, Mignone F, Saggese I, Guarino D, Costella A, Molinario R, De Bonis M, Ferrandina G, Petrillo M, Scaglione GL, Capoluongo E. *Clinical impact on ovarian cancer patients of massive parallel sequencing for BRCA mutation detection: the experience at Gemelli hospital and a literature review.* Expert Rev Mol Diagn. 2015; 15(10):1383-403.

[17] Stenson PD, Mort M, Ball EV, Shaw K, Phillips A, Cooper DN. *The Human Gene Mutation Database: building a comprehensive mutation repository for clinical and molecular genetics, diagnostic testing and personalized genomic medicine.* Hum Genet. 2014 Jan;133(1):1-9.

[18] Rehm HL, Bale SJ, Bayrak-Toydemir P. et al. *ACMG clinical laboratory standards for next-generation sequencing.* Genet Med 2015;15(9):733-47.

[19] Schouten JP, McElgunn CJ, Waaijer R, Zwijnenburg D, Diepvens F, Pals G. *Relative quantification of 40 nucleic acid sequences by multiplex ligation-dependent probe amplification.* Nucleic Acids Res. 2002; 30(12):e57.

[20] Qiu P, Stevens R, Wei B, Lahser F, Howe AYM, et al. *HCV Genotyping from NGS Short Reads and Its Application in Genotype Detection from HCV Mixed Infected Plasma.* PLOS ONE 2015; 10(4): e0122082.

[21] Prabhav Kalaghatgi, Anna Maria Sikorski, Elena Knops, Daniel Rupp, Saleta Sierra, Eva Heger, Maria Neumann-Fraune, Bastian Beggel, Andreas Walker, Jörg Timm, Hauke Walter, Martin Obermeier, Rolf Kaiser, Ralf Bartenschlager and Thomas Lengauer. *Geno2pheno[HCV] – A Web-based Interpretation System to Support Hepatitis C Treatment Decisions in the Era of Direct-Acting Antiviral Agents.* PLoS One 2016; 11(5): e0155869.

[22] Lontok E, Harrington P, Howe A, Kieffer T, Lennerstrand J, Lenz O, McPhee F, Mo H, Parkin N, Pilot-Matias T, Miller V. *Hepatitis C virus drug resistance-associated substitutions: State of the art summary.* Hepatology. 2015 Nov; 62(5):1623-32.

[23] Jane P Messina, Isla Humphreys, Abraham Flaxman, Anthony Brown, Graham S Cooke, Oliver G Pybus, and Eleanor Barnes. *Global Distribution and Prevalence of Hepatitis C Virus Genotypes.* Hepatology. 2015 Jan; 61(1): 77–87.

[24] Tellinghuisen TL, Foss KL, Treadaway J. *Regulation of hepatitis C virion production via phosphorylation of the NS5A protein.* PLoS Pathog 2008; 4: e1000032.

[25] Bartenschlager R, Lohmann V, *Replication of hepatitis C virus.* J Gen Virol 2000; 81:1631-1648.

[26] Lohmann V, Roos A, Korner F, Koch JO, Bartenschlager R. *Biochemical and structural analysis of the NS5B RNA-dependent RNA polymerase of the hepatitis C virus.* J Viral Hepat 2000; 7:167-174.

Chapter 4

Algorithms for NGS

RNA-seq is a set of powerful techniques that allow to characterise metabolic activity of a biological sample by sequencing all of its mRNA [1].
Unfortunately, as seen in chapter 1 the output of a NGS sequencing platform is a vast set of short reads that have to be properly assembled in order to reconstruct original transcripts.
When high quality reference sequences are available for the organism that is object of study, reconstruction can be performed by alignment to reference genome for which many efficient and accurate solutions are available.
In case references are unavailable *denovo assembly* [2] has to be applied: reads are assembled blindly in order to reconstruct transcripts.

Currently available tools for denovo assembly of RNA-seq data - such as Bridger [3], Oases [4] and Trinity [5] - achieve high levels of sensitivity and share a similar approach as they rely on the identification of k-mer sequences. Bridger then uses this information to build and traverse splicing graphs [6], while Oases and Trinity rely on De-Bruijn graphs [7].
Despite the good sensitivity, they all show two main limitations that will be further discussed with results:

- High number of false positive reconstructions
- Very high demands in term of hardware requirements

When working with real data, in absence of any reference it is not trivial - and maybe not even possibile - to determine the correctness of a reconstruction, so applying methods that are likely to produce many false positives can lead to production of unreliable results.
Furthermore current approaches are very demanding in terms of hardware requirements: this can be a serious limitation, especially in the case of smaller laboratories where dedicated computing infrastructures are not available.

To overcome these limitations we developed **STAble** [8], a new denovo assembler for RNA-seq data that is built around an original approach: the whole reads are used to drive the alignment process instead of considering only smaller k-mers with the aim of reducing the number of false positive reconstructions.
False positive matter was our primary concern, however it is desirable to achieve a sensitivity at least comparable to existing tools.
Moreover we designed it to be parallelizable, allowing to to split the assembly process into smaller subtasks that can be processed in parallel even in absence of dedicated computing infrastructures.

STAble's performances were initially compared with other tools on many simulated and real RNA-seq datasets, but given the growing interest on metagenomics analysis we also explored the possibility of applying it for the assembly of metatranscriptome.
Metatranscriptome studies are still at an early stage and identification of expressed transcripts is usually accomplished by alignment of short reads (obtained by *Whole-Transcriptome shotgun sequencing (WTSS)* [9] as for RNA-seq) to databases of annotated bacterial transcripts, that are later mapped to known metabolic pathways in order to characterise metabolic activity.

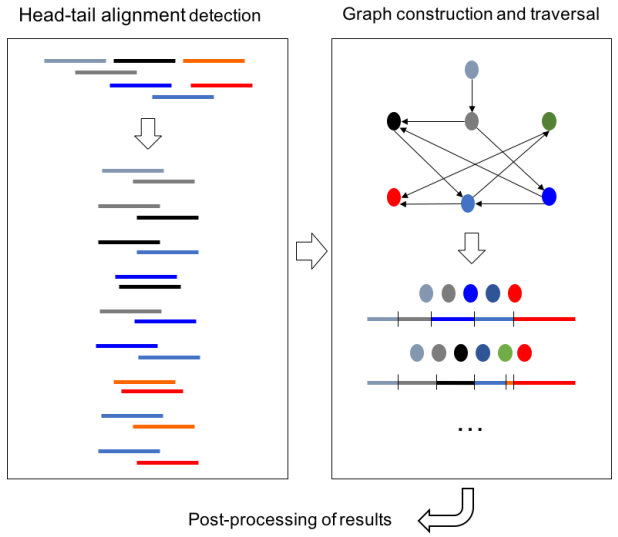

Figure 4.1: STAble's assembly workflow.

This approach is however expected, given to short length of reads, to produce many assignment ambiguities and novel unannotated transcripts can be missed.
For these reasons we decided to tune STAble for this application: preliminary results are presented in section 4.2.

4.1 STAble

STAble's assembly workflow consists of three modules as depicted in Figure 4.1.

The first step is the efficient detection of potential head-tail alignments between reads, possibly with mismatches: ideally if two reads show an head-tail overlap with a good score they can be assembled into a longer contig.
Second module uses reported alignments to build an unweighted directed graph which is traversed by a custom algorithm that takes into account biological properties of input data.

Finally the third one applies some post-processing operations to results.
Core analysis modules are implemented in C while the rest of the pipeline is written in Perl.

4.1.1 Head-tail alignments detection

The first module starts from a FASTQ file with raw sequencer output on alphabet ACGTN and returns a list of triples $[i, j, k]$ where:

- i and j represent two reads.
- k is an integer number.
- The tail of i (that is $i[k...len(i)]$) has a good overlap with the head of j (that is $j[1...(len(i) - k + 1)]$).

Given this definition, it is vital to explain what is considered to be a *good overlap*, a concept that must take into account biological properties of input data.
From a pure computer science perspective there are two main approaches to string alignment:

- **Exact alignment:** a potential alignment is valid only if *edit distance* is 0.
- **Alignment with errors:** alignment is valid if edit distance is included between 0 and a threshold t.

For this specific application the presence of errors must be tolerated to deal with possible sequencing errors. More formally, an head-tail alignment is reported as valid only if it satisfies the following conditions:

1. Edit distance must not be greater than max_errors.

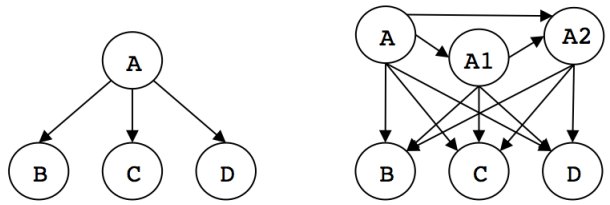

Figure 4.2: The graph on the left represents a situation where sequence A can be extended with B, C and D. Due to high sequencing depths the existence of $A1$ and $A2$ that are very similar to A is highly probable. If a maximum overlap length is not set, the situation on the right is likely to occur: the graph will contain many paths that will complicate its structure but do not contribute with meaningful informations.

2. Overlap length is included in interval $[min\_len, max\_len]$.

The first condition trivially ensures the goodness of the alignment score: for increased efficiency only mismatches are considered.
A minimum length for the overlap is required to avoid alignments potentially caused by casual similarities. This is particularly important for dealing with low complexity and repeated regions that are likely to produce many chimeric contigs if too short overlaps are accepted.
Similarly a maximum length must be set to deal with redundancy of information caused by high sequencing depths: an alignment caused by an excessive overlap will generate a contig just a little longer than a single read, while there are good chances that the same reads can generate more informative alignments elsewhere. In addition this threshold helps to reduce the number of arcs in the graph built by the second module, preventing the formation of alternative paths that represent unessential variants of the same transcripts as shown in Figure 4.2. Always referring to the same figure it is important to point out that even if alignments between A, $A1$ and $A2$ are prevented, their tails still align with the same sequences: to address this, when a too similar sequence is detected it is discarded.

In addition there is also a limit to the number of alignments a single read can contribute to on both head and tail. The heuristic is that if alignment $[i, j, k]$ is discarded because read i was used too many times, it is highly probable that a very similar read i' exists and makes the alignment $[i', j, k']$ valid too. Without this limitation, for each alignment $[i, j, k]$ and a generic pair of sequences i' and j' (very similar to i and j) any possible combination such as $[i', j, k']$, $[i, j', k'']$ and $[i', j', k''']$ is likely to be considered, but once again this would just result in a more complex graph without additional benefits.
The proposed algorithm for head-tail alignment detection will now be examined, starting with the definition of keywords, parameters and data structures involved.

Keywords

Anchor A k-mer used to start an alignment, whose length is defined by an input parameter. An anchor is valid only if it contains all four nucleotides, in order to filter low complexity regions.

Parameters

anchor_size Length of the anchors expressed in nucleotides (default: 11).

anchor_scope Maximum number of anchors for each sequence to consider for starting alignment. Its meaning will become more clear during algorithm presentation (default: 5).

max_mismatches Maximum number of mismatches allowed (default: 10% of overlap length).

min_overlap_len Minimum length allowed for overlaps (default: 20% of longer sequence[1]).

max_overlap_len Maximum length allowed for overlaps (default: 90% of shorter sequence).

max_align_before Maximum number of alignments the head of a sequence can contribute to (default: 5).

max_align_after Maximum number of alignments the tail of a sequence can contribute to (default: 5).

Data structures

Anchor index This is the main data structure for this module, depicted in Figure 4.3. It is implemented as an array whose indexes range across binary representations of all possible anchors and is used to keep record of all occurrences of each anchor in input reads.
Lists of occurrences are implemented as dynamically allocated linked lists, since a sequential visit is required by the algorithm. Each list entry contains an integer identifier for the sequence and the offset of the occurrence relative to its start.

Algorithm description

The pseudocode of the main procedure - informally named **SeqAlign** - is represented in Algorithm 1. Computation starts by recoding input FASTQ from 8-bit ASCII characters to a 2-bit alphabet as shown in Figure 4.4: this will allow to reduce memory consumption and to speed up subsequent operations.
No special symbol is assigned to ambiguous bases - such as N - but the

[1]RNA-seq reads are expected to have all the same length, however the algorithm is designed to work even for sequences with different lengths.

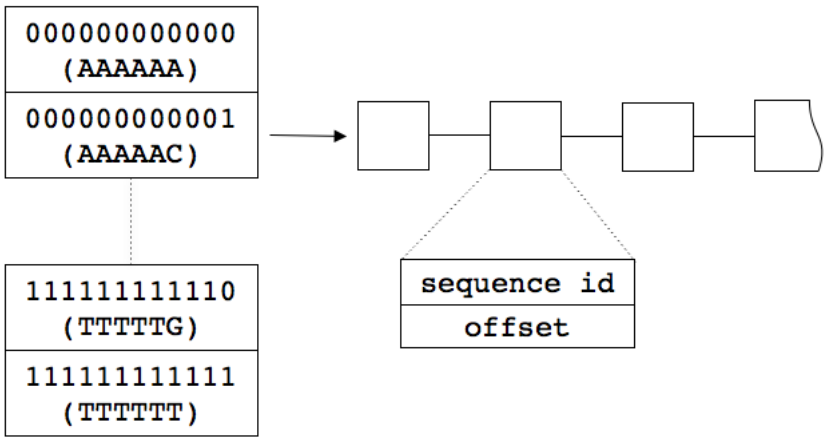

Figure 4.3: Representation of the **anchor index** data structure for *anchor_size=6*, supposing that each base is encoded with a 2-bit simbol: A=00, C=01, G=10, T=11.

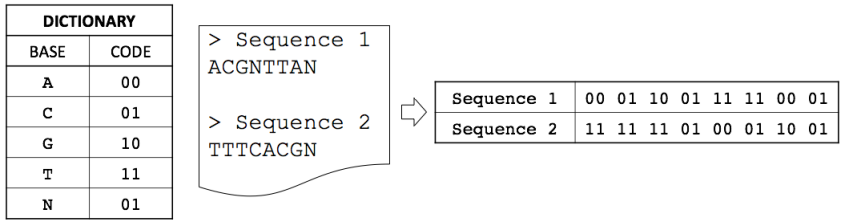

Figure 4.4: Encoding of input FASTQ on a 2-bit symbols alphabet. Ambiguous bases are encoded with the same symbol reserved for C.

Algorithm 1: SeqAlign

Input: fa_input (FASTQ sequences)
Output: A set of triples $[i, j, k]$, each one representing and head-tail alignment between i and j starting from position k in i.

```
   // Encodes input on a 2-bit alphabet.
1  seq_arr ← encode_fastq(fa_input);
   // Initialise output set and anchor index.
2  alignments ← ∅;
3  anchor_index ← ∅;
4  foreach s ∈ seq_arr do
       // Initialise the list of sequences already aligned with s.
5      visited ← ∅;
6      end ← lenght(s) − anchor_size;
7      foreach idx ∈ [0 ... end] do
           // Reads next anchor from s.
8          anchor = read_sequence(s, idx, idx + anchor_size);
           // Add the new occurrence to index.
9          anchor_index[anchor] ← anchor_index[anchor] ∪ [s, idx];
           // Process only first and last anchor_scope anchors.
10         if idx ∈ [0 ... anchor_scope] ∪ [end − anchor_scope ... end] then
               // FindAlignment returns a triple [i,j,k] that
                  represents an alignment.
11             al ← FindAlignment(anchor, s, idx, visited);
12             alignments ← alignments ∪ al
               // Verify if acceptance s was discarded during
                  alignment process.
13             if is_discarded(s) then
14                 break;
15             end
16         end
17     end
18 end
19 return alignments;
```

same symbol reserved for C is used. This choice was made to keep the size of the new alphabet as low as possible. Results quality is not affected since reads with too many ambiguous base are usually discarded by pre-processing steps because of low quality, so false matches with C are expected to be rare. The use of C was chosen because in biological data poly-A and poly-T stretches are very frequent, so that could have caused much more problems.
After initialisation is done, the algorithms proceeds to analyse input sequences one at a time. For each one the *visited* set is kept (implemented as a linked list): it contains ids of the sequences that have been successfully aligned before with current one. This is an effective strategy for avoiding multiple alignments between the same sequences: they would be collapsed anyway during graph construction step (as they would result in the same arc), however this way there is a time saving by avoiding an useless computation.
Each read is examined anchor by anchor and they are all indexed. For the first and last *anchor_scope* anchors additional operations are performed:

1. Current anchor is given as input to **FindAlignment** procedure (Algorithm 2) that will try to detect potential alignments.

2. If a good alignment is found it is added to report list.

3. Check if alignment procedure caused current sequence to be discarded (because it is too similar to an already processed one or *max_align_before* and *max_align_after* limits are exceeded).

anchor_scope limit was introduced to cut down complexity: if two sequences can be successfully head-tail aligned, one of the first (or last) anchors must match.

The first operation performed by FindAlignment is to query anchor

Algorithm 2: FindAlignment

Input: *anchor* (current anchor), *cur_id* (current sequence id), *anchor_start* (anchor offset in current sequence), *visited* (a set with ids of sequences already aligned with current one).

Output: A triple $[i, j, k]$ representing an head-tail alignment between i and j starting from offset k in i.

```
// Retrieves previous occurrences of same anchor from index.
occs[] ← anchor_index[anchor];
foreach o ∈ occs do
    // Ignore previous occurrences on the same read.
    if cur_id = o.seq_id then
        next;
    end
    // Candidate already aligned with current one or
       max_align_before and max_align_after are exceeded.
    if cannot_align(cur_id, o.seq_id, visited) then
        next;
    end
    // ov represents start and end index of potential overlap
       area.
    ov ← get_overlap(cur_id, o, anchor_start);
    // Overlap too short or one sequence included in the other.
    if illegal_overlap(ov) then
        next;
    end
    mismatches ← CompareSequences(cur_id, o.seq_id, ov)
    if mismatches ≤ max_mismatches then
        // Discard current one if too similar to candidate.
        if ov.length > max_overlap_len then
            mark_as_discarded(cur_id);
            return ∅;
        end
        // cur_id is before o.seq_id in alignment.
        if is_before(cur_id, o.seq_id, ov) then
            first ← cur_id; last ← o.seq_id; k ← ov.start1;
        else
            first ← o.seq_id; last ← cur_id; k ← ov.start2;
        end
        first.aligned_after ← first.aligned_after + 1;
        last.aligned_before ← last.aligned_before + 1;
        visited ← visited + o.seq_id;
        return [first, last, k];
    end
end
return ∅;
```

Figure 4.5: During alignment process reads are overlapped on a specific anchor to start alignment.

index for past occurrences of input anchor. For each one three actions are taken:

1. Preliminary checks
2. Overlap area detection
3. Alignment scoring

During step 1, involved sequences are checked for trivial incompatibilities:

- The two occurrences must not refer to current sequence: such alignment would make no sense.
- Candidate sequence must not have been aligned with current one before or marked as *discarded* (because too similar to another one or because of *max_align_before* and *max_align_after* limits.)

On the second step, sequences are overlapped on anchors to identify potential overlap area as shown in Figure 4.5. In case its extension is smaller than *min_overlap_len* or the alignment results in the inclusion of one sequence into the other (this situation occurs in case of input sequences with different lengths), procedure skips to next occurrence.

Finally in step 3 the alignment score is computed by **Compare-Sequences** (Algorithm 3), that efficiently computes number of mismatches on overlap area by using XOR metric.

Algorithm 3: CompareSequences

Input: $s1$, $s2$ (ids of sequences to compare), ov (index that identify overlap area).
Output: Number m of mismatches in overlap area.

```
// Reads data from sequences.
```
1 $data1 = read\_sequence(s1, ov.start1, ov.end1)$;
2 $data2 = read\_sequence(s2, ov.start2, ov.end2)$;
```
// Computes bitwise XOR.
```
3 $mm = data1 \oplus data2$;
```
// Returns the number of bit pairs greater than 0.
```
4 **return** $countMismatches(mm)$;

Given that $x \oplus y = 1 \Leftrightarrow x \neq y$, computation of $s1 \oplus s2$ results on a third sequence r for which the following property holds:

$$\forall i : r[i] \neq 0 \Leftrightarrow s1[i] \neq s2[i] \tag{4.1}$$

Since input is encoded in 2-bit symbols the number of mismatches can be easily calculated by applying XOR operator and counting number of bit pairs with value greater than 0.
If return value is lower than *max_mismatches*, FindAlignment performs two additional steps:

- If overlap length is longer than *max_overlap_length* current sequence is marked as discarded because too similar to the candidate.

- If all constraints are satisfied, it is established which read is the head and which the tail, updating respective *aligned_before* and *aligned_after* counters afterwards.

Finally, if alignment has not been discarded yet it is reported since it has passed all filters.

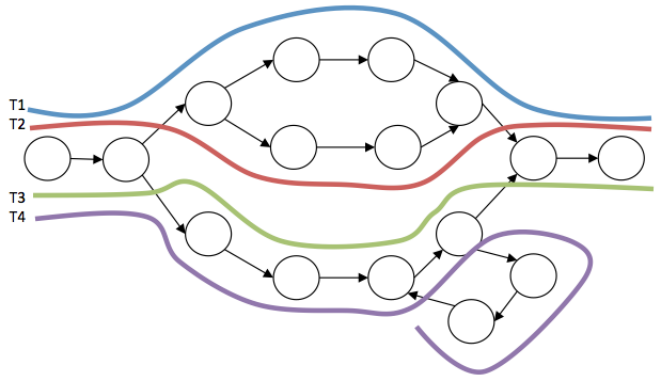

Figure 4.6: Distinction between "big" and "small" bubbles. T1 and T3 are both to be reported since we consider them as alternative splicing forms of the same gene. T2 is to be discarded, as it is considered as a slight variant of T1. T4 is to be reported too, but marked as "low confidence" since it does not end on a sink.

4.1.2 Graph construction and traversal

The second module uses reported head-tail alignments to build an unweighted directed graph where each node represents a read and each arc an head-tail alignment between two of them.
Ideally every path originating in a source and ending into a sink represents a transcript or a fragment of it. However in real world cases this is not true due to high sequencing depths, the presence of alternative splicings and the head-tail alignments on repeated regions that may lead to chimeric reconstructions: this graph must be carefully traversed taking into account all these issues.

A first set of expected situations is depicted in Figure 4.6. We consider T1 and T2 to be two alternative paths that represent the same biological sequence and we want to report only one of them. This is because of the presence of what we informally call a *small bubble*: a single path is bifurcated in two or more parallel ones that merge again after few nodes. We consider this class of *bubbles* to represent a consequence of high sequencing depths and consequent redundancy

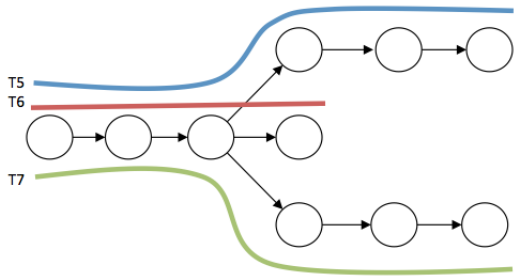

Figure 4.7: In this scenario only T5 and T7 are to be reported because T6 is not considered as a significant variant.

of information.
T3 instead is part of a *big bubble* with T1 so we consider these transcripts as alternative splicing forms of the same gene and we want to report both.
Finally even T4 is due to be reported, however since it is the result of a loop cut (it does not end on a sink) it will be marked as *low confidence*: this way the user is informed that is derived from a particular situation and should be used carefully. The distinction between "big" and "small" bubbles is made relying on a threshold expressed as a number of nodes.

Another possible scenario is shown in Figure 4.7. A path branches off in several directions that do not merge anymore, so we have a common source but different tails.
In this case we define a threshold on the length of the tail: only significantly different variants are to be reported, the others are labeled as *short tendrils* and ignored.
The same criteria applies for the reversed situation when there are multiple sources that merge in the same branch: a minimum number of original nodes in the head is required.

Parameters

intT The number of nodes used to discriminate between "big" and "small" bubbles, referring to situation in Figure 4.6 (default: 5).

extT With reference to Figure 4.7, this is the number of original nodes required in the head or tail (default: 2).

minL Minimum length for paths to be reported (default: 2).

Algorithm description

The second module procedure - informally named **cpaths** - is presented in Algorithm 4. It consists of three different steps:

1. An unweighted directed graph is built from SeqAlign results: each node represents a read and each arc an head-tail overlap.

2. Graph is searched for *short tendrils* that are removed.

3. Graph is traversed to detect all paths that satisfy all our requirements.

Algorithm 4: CPATHS

Input: sa_res (the list of triples returned by SeqAlign), $intT$, $extT$.
Output: A list of paths in a graph built from SeqAlign results.

```
// Build graph from SeqAlign results.
```
1 $g \leftarrow buildGraph(sa\_res)$;
```
// Prune short tendrils.
```
2 $g \leftarrow pruneTendrils(g)$;
```
// Traverse graph to build paths.
```
3 $paths \leftarrow traverseGraph(g)$;
4 **return** $paths$;

Operations performed by the first step are trivial: every input triplet $[i, j, k]$ is translated by adding arc $i \rightarrow j$ to the graph.

Second step consists in pruning of short tendrils by using input parameter $extT$ as a threshold.
For each node n the algorithm verifies the existence of paths that start in that node and have length greater than $extT$:

- If at least one exists, all outgoing arcs that lead to paths shorter than the set threshold are removed.
- If not, on the outgoing arcs leading to paths of maximum length are kept.

The same logic is then applied by traversing arcs backwards: for each node n we want to assess the presence of incoming paths longer than $extT$.
At this point pruning of short tendrils is completed.

The third and last step represents the core of the traversal algorithm. It is based on the principles of *depth-first search (DFS)* and starting from the pruned graph detects all paths longer than $minL$ after taking care of all big and small bubbles issues.
The algorithm is very complex since it has to deal with a vast number of situations, so instead of showing and intricate pseudocode, a more intuitive high-level description will be given.
DFS is performed from all sources s, then for each node n that is found the first operation is to check if it is a sink.
In affermative case, if the path from s to n is at least $minL$ long, it is added to result set. If not every child m of n is examined:

- If m is already in the path from s to n we have a loop, so the arc that causes it is cut and the path is reported but in the *low confidence* group.
- If path from s to n contains a node w such that the sub-path from w and m is shorter than $intT$ and in previously reported paths there is another one that contains a sub-path from w to m

also shorter that $intT$ we have detected a small bubble, so the loop continues and next child of n is examined.

- If none of the previous conditions is verified DFS continues by travelling through m.

DFS approach was chosen because of its memory efficiency, that is desirable given that typical graphs are composed by millions of nodes. On the other hand, from a time perspective it is less effective since in presence of nodes with $indegree > 1$ multiple visit to the same subgraph are performed.
In order to avoid this we applied an heuristic to speed up the process. Let's suppose that node n has $indegree > 1$ and arc $p \rightarrow n$ was already travelled: when DFS ascends to n subgraph originating from it is fully explored and all valid sub-paths are already detected. For each node with multiple parents we keep track of all its valid subpaths, so when a new arc $q \rightarrow n$ is travelled, multiple paths are immediately reported by appending all possible continuations to current one without performing a new visit. When node n has been visited a number of times equals to its indegree the list of sub-paths is deleted to free memory.
This technique allows to visit each node only once, but it may also generate many paths that are very similar to each other: to address this we introduced the $whiL$ threshold.
At the beginning all nodes are coloured of *white* and every time a path is reported, all of its nodes are coloured of *black*. A path is reported only if it contains at least $whiL$ white nodes: that means it will introduce new significative information compared to the already reported ones.

From a theoretical perspective, if the graph contains an exponential number of paths that satisfy requirements the traversal operation has exponential time complexity. However, due to filtering criteria ap-

plied in first module, in real data experiments this condition has never occurred and visit always completed within few seconds in graphs with millions of nodes.

4.1.3 Results post-processing

The first operation performed by this module is the conversion of paths reported by second module to a FASTA file: SeqAlign output and input FASTQ contains all the informations required to accomplish this task.
Then the clustering algorithm implemented by **vsearch** [10] is applied to resulting file: due to high redundancy of NGS data it is possible that different paths that are completely unrelated in terms of graph nodes represent the same biological sequence, so they are collapsed.

4.1.4 Parallel workflow

STAble is designed to be parallelizable, allowing to break down computations on huge datasets into multiple smaller subtasks that can be individually run even in absence of dedicated computing infrastructures.
The proposed workflow is depicted in Figure 4.8. Input dataset is split in blocks of size K each (a typical value is 1 million reads) that will be processed in parallel.
Once all computations are completed, partial results are collected and used as input for a new iteration. When size of resulting dataset is smaller than K, a last iteration takes place and final results are returned.

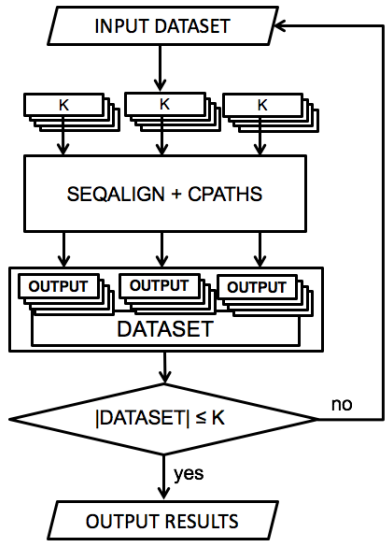

Figure 4.8: STAble's parallel workflow.

4.2 Benchmarks

STAble's performances were evaluated and compared with Oases and Trinity - that currently represent the most widely used state-of-the-art tools for denovo assembly - and with Bridger, which is the most recently introduced one.

4.2.1 Benchmarks on simulated datasets

STAble's performances, in a preliminary stage were tested on a large set of simulated datasets.
While benchmarks are usually performed on real data for which a reference genome is available, we have chosen to work with simulations because they allow the unambiguous identification of what is true and false.
With real data the correctness of reconstructions is usually assessed by alignment to genome or to a database of known transcripts, so as

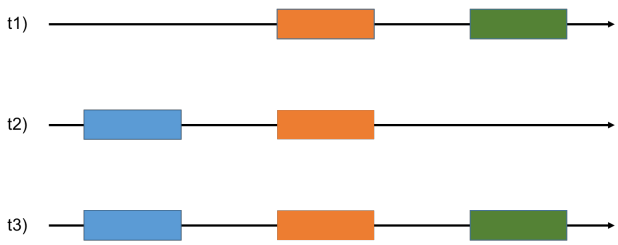

Figure 4.9: Let's suppose that t1, t2 and t3 are three alternative splicing forms of the same gene and that only t1 and t2 are present in the sample: reads may lead to reconstruction of t3 even if it is not effectively present.

long as reconstructions are compatible with genome are considered as true.

With simulated datasets instead, we are able to apply more strict filters by accepting only those transcripts that were effectively present in database used for simulation: this opportunity allowed us to characterise a new and subtle type of false positive.

Figure 4.9 represents three different splicing forms of the same gene. Let's suppose that only t1 and t2 are effectively expressed in sequenced sample: assembly process may lead to reconstruction of t3 even if it is not present.

With simulated datasets we have a full knowledge of what we expect to reconstruct, so we can easily label t3 as a false positive and try tuning the assembler not to produce it.

On real data scenario instead, t3 is compatible with genome and corresponds to a known annotated transcript: it must be accepted as true with a consequent underestimation of real false positive ratio.

To deal with this situation, in following benchmarks two classes of false positive are defined:

- **False positive class A (FPA):** reconstructions that are compatible with genome but do not correspond to any of the transcripts used to perform sequencing simulation.

- **False positive class B (FPB):** chimeric reconstructions that are not compatible with genome.

Simulated benchmarks setup

Input reads have been simulated using ART [11] as Illumina 150bp single end with 20x of fold coverage and HiSeq 2500 quality profile.
Reconstructions were validated by alignment to database used for simulation using BLASTN [12]: only reconstructed transcripts that are fully included in one of the reference sequences are accepted and reported as true positives.
False positives are then aligned to genome with GMAP [13]: compatible ones are labeled as FPA and the others as FPB.
A reference sequence that has been reconstructed for at least 90% of its length is labeled as full-length reconstructed.
STAble ran on a small grid of 9 computers equipped with Intel(R) Core(TM) i3-2130 CPU @ 3.40GHz processor and 8 GB of RAM, while other assemblers were executed with default parameters on a server equipped with Intel(R) Xeon(R) CPU E5-2665 @ 2.40GHz processor and 48 GB of RAM.

Simulated datasets

Results for 4 simulated datasets will be shown. Size of these samples is much smaller than common datasets: they were chosen to highlight the key points of strength of the new strategy, but also results on full-size real datasets will be presented afterwards.

Dataset A 147800 reads simulated from a pool of 200 transcripts randomly chosen from human transcriptome.

Dataset B 1088271 reads simulated from a pool of 6309 transcripts randomly chosen from human transcriptome.

Dataset C 1242040 reads simulated from transcriptome of 10 randomly chosen bacterial species (11815 total transcripts).

Dataset D 2382790 reads simulated from transcriptome of 50 randomly chosen bacterial species (43578 total transcripts)

Abbreviations

- **Assembler** Assembler name.
- **# of results** Total number of reported reconstructions.
- **# of FP** Total number of false positive reconstructions (FPA+FPB).
- **FPA** Number of false positives classified as FPA.
- **FPB** Number of false positives classified as FPB.
- **100%** Number of reference sequences reconstructed at full-length.
- **80%** Number of reference sequences reconstructed at least at 80% of length.
- **70%** Number of reference sequences reconstructed at least at 70% of length.
- **S100** Sensitivity value for full-length reconstructions.
- **S80** Sensitivity value for reconstructions of at least 80% of length.
- **S70** Sensitivity value for reconstructions of at least 70% of length.
- **FPR** Global false positive ratio (FPA+FPB).

Dataset A Dataset A consists of 147800 reads simulated from a pool of 200 transcripts randomly chosen from human transcriptome (see Table 4.1).
STAble showed a sensitivity similar to other assemblers while producing only 3 false positives. It is interesting to note that Oases showed the highest sensitivity but also the highest number of false positives.

Table 4.1: Dataset A - 200 random human transcripts. STAble returned the most reliable set of results showing a sensitivity comparable to other assemblers while producing only 3 false positives.

Assembler	# of results	# of FP	FPA	FPB	100%	80%	70%	S100	S80	S70	FPR
STAble	249	3	1	2	156	163	166	78%	82%	83%	1%
Bridger	210	61	15	46	140	144	145	70%	72%	73%	29%
Oases	321	114	45	69	158	164	164	79%	82%	82%	36%
Trinity	258	59	29	30	157	164	167	79%	82%	84%	22%

Dataset B Dataset B consists of 1088271 reads simulated from a pool of 6309 transcripts randomly chosen from human transcriptome (see Table 4.2).
Even in this scenario STAble produced the lowest number of false positives while exhibiting a sensitivity comparable to other tools.
While Oases and Trinity showed a slightly higher number of transcripts reconstructed at 80% and 100% it is important to point out that they also show the highest rate of false positives.
While false reconstructions produced by Bridger and Oases are mostly FPB, Trinity showed a very high number of FPA: if this was a benchmark on real data (where it is not possible to discriminate between the two classes because the set of effectively expressed sequences is not available) it would have shown a way better performance.
Moreover when considering reference transcripts reconstructed at at least 70% STAble performances are almost the same as Trinity's.

Table 4.2: Dataset B - 6309 random human transcripts. Although STAble's sensitivity is a bit lower than Oases' and Trinity's, results reliability is still the best.

Assembler	# of results	# of FP	FPA	FPB	100%	80%	70%	S100	S80	S70	FPR
STAble	9071	1373	551	822	2682	3417	4206	43%	54%	67%	15%
Bridger	5697	1957	454	1503	2654	3044	3189	42%	48%	51%	34%
Oases	16895	6334	1075	5259	3484	3926	4067	55%	63%	65%	37%
Trinity	8300	2665	1771	894	3536	4023	4236	56%	64%	67%	32%

Dataset C Dataset C consists of 1242040 reads simulated from transcriptome of 10 randomly chosen bacterial species for a total of 11815 transcripts (see Table 4.3).
STAble showed the highest sensitivity while minimising false positive ratio. It is interesting to note that due to absence of alternative splicings in bacterial transcriptome it is not possible to produce FPA.

Table 4.3: Dataset C - 11815 mixed bacterial transcripts. STAble showed the best sensitivity while producing the lowest false positive ratio alongside with Trinity.

Assembler	# of results	# of FP	FPA	FPB	100%	80%	70%	S100	S80	S70	FPR
STAble	18218	454	0	454	9661	10128	10218	82%	86%	87%	2%
Bridger	5873	302	0	302	8437	8877	8971	71%	75%	76%	5%
Oases	5579	284	0	284	6633	8207	8523	56%	69%	72%	5%
Trinity	7597	157	0	157	9039	9342	9471	77%	79%	80%	2%

Dataset D Dataset D consists of 2382790 reads simulated from transcriptome of 50 randomly chosen bacterial species for a total of 43578 transcripts (see Table 4.4).
These results highlight an important feature of STAble: running on a small grid of desktop computers equipped with just 8GB of RAM, it was the only assembler capable of completing the assembly task. All existing tools terminated returning an *out of memory* error even on a

server with 48GB of RAM.

Table 4.4: Dataset D - 43578 mixed bacterial transcripts. Existing assemblers terminated with an *out of memory* error on a computer with 48GB of RAM. STAble ran on a computer with just 8GB of RAM.

Assembler	# of results	# of FP	FPA	FPB	100%	80%	70%	S100	S80	S70	FPR
STAble	71159	1531	1531	0	29961	36547	37912	68%	84%	87%	2%

4.2.2 Benchmarks on real data

STAble's performances were compared with Trinity and Oases on two real datasets.
Since we were not able to satisfy their hardware requirements on our servers with real data, we decided to perform comparison using benchmark datasets from their respective papers (SRX017794 for Oases and GSE29209 for Trinity) and to rely on published results.
STAble ran on a small grid of 9 computers equipped with Intel(R) Core(TM) i3-2130 CPU @ 3.40GHz processor and 8 GB of RAM.
In these scenarios it is not possible to discriminate between different classes of false positives, so the total amount will be used.

STAble vs Oases

STAble and Oases were compared on mouse dataset SRX017794 (about 84M of paired-end Illumina reads). Results are reported in Table 4.5. STAble achieved a sensitivity comparable to Oases while producing a significantly lower number of both reported transcripts and false positives.
From the hardware requirements perspective, Oases paper states that assembly process took about 10 hours on a server equipped with a 48

Table 4.5: STAble vs Oases on mouse dataset SRX017794. Results reported for Oases are from Table 2 and Supplemental Table 6 of [4].

Assembler	# of results	# of FP	100%	FPR
STAble	76396	2378	1332	3%
Oases	175914	29906	1324	17%

core AMD Operon processor and 256GB of RAM while in our grid of 9 common desktop computers the task was completed in 8 hours.

STAble vs Trinity

STAble and Trinity were compared on mouse dataset GSE29209 (about 105M of paired-end Illumina reads). Results are reported in Table 4.6.

Table 4.6: STAble vs Trinity on mouse dataset GSE29209. Results reported for Trinity are from Supplementary Table 2 of [5].

Assembler	# of results	# of FP	100%	FPR
STAble	207349	12047	11711	6%
Trinity	179340	147634	11334	82%

Even in this test scenario STAble showed a sensitivity comparable to Trinity while producing a drastically lower number of false positive reconstructions.
Analysis with Trinity took 60 hours on a server equipped with 256 GB of RAM and a *load sharing facility* that ran some steps in parallel. STAble completed the task in just 7 hours on our test environment.

Bibliography

[1] Zhong Wang, Mark Gerstein & Michael Snyder. *RNA-Seq: a revolutionary tool for transcriptomics.* Nature Reviews Genetics 2009; 10, 57-63.

[2] Gordon Robertson et al. *De novo assembly and analysis of RNA-seq data.* Nature Methods 7, 2010; 909–912.

[3] Chang Z, Li G, Liu J, Zhang Y, Ashby C, Liu D, Cramer CL, Huang X. *Bridger: a new framework for de novo transcriptome assembly using RNA-seq data.* Genome Biology 2015; 16:30.

[4] Schulz MH, Zerbino DR, Vingron M, Birney E. *Oases: robust de novo RNA-seq assembly across the dynamic range of expression levels.* Bioinformatics 2012; 28: 1086–1092.

[5] Grabherr MG, Haas BJ, Yassour M, Levin JZ, Thompson DA, Amit I, et al. *Full-length transcriptome assembly from RNA-Seq data without a reference genome.* Nat Biotechnol. 2011; 29: 644–652.

[6] Heber S, Alekseyev M, Sze SH, Tang H, Pevzner PA. *Splicing graphs and EST assembly problem.* Bioinformatics. 2002;18 Suppl 1:S181-8.

[7] Phillip E C Compeau, Pavel A Pevzner & Glenn Tesler. *How to apply de Bruijn graphs to genome assembly.* Nature Biotechnology 2011; 29, 987–991.

[8] Igor Saggese, Giovanni Manzini and Flavio Mignone. *STAble: a novel approach to denovo assembly of RNA-seq data.* 13th International Meeting CIBB 2016; 2016 Sept 1-3; Stirling - UK.

[9] In Seok Yang and Sangwoo Kim *Analysis of Whole Transcriptome Sequencing Data: Workflow and Software.* Genomics Inform. 2015 Dec; 13(4): 119–125.

[10] Torbjørn Rognes, Tomáš Flouri, Ben Nichols, Christopher Quince and Frédéric Mahé. *VSEARCH: a versatile open source tool for metagenomics.* PeerJ. 2016; 4: e2584.

[11] Weichun Huang, Leping Li, Jason R Myers, and Gabor T Marth. *ART: a next-generation sequencing read simulator.* Bioinformatics 2012; 28 (4): 593-594.

[12] Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. *Basic local alignment search tool.* J Mol Biol. 1990 Oct 5; 215(3):403-10.

[13] Wu TD, Watanabe CK. *GMAP: a genomic mapping and alignment program for mRNA and EST sequences.* Bioinformatics. 2005 May 1;21(9):1859-75.

Chapter 5

Discussion

The advent of NGS techniques revolutionised genomic research but for clinical routine purposes there are still some open challenges, especially on the bioinformatics side, that are limiting their application as a support for traditional Sanger sequencing.
In this thesis a new workflow for variant calling and annotation starting from NGS data was introduced. It is designed to be flexible and by setting up a proper analysis profile it is applicable to study any annotated gene in combination with all the major sequencing platforms available.
The entire procedure was extensively validated and as an added value from the state-of-the-art it provides automatic integration with external resources (like databases of clinically relevant mutations and functional predictors) by automatically resolving problems caused by the use of different standards for variant nomenclature.
Users can also choose their preferred visualisation format and particular attention was paid to HGVS recommendations [1] that are offered as default.
This new workflow was used as core for the development of two integrated tools: Amplicon Suite and SmartVir.

Amplicon Suite is a platform for variant calling and annotation:

by setting up a proper profile it is possible to analyse NGS amplicon sequencing data from any annotated gene.
In collaboration with research group of Dr. Ettore Capoluongo of Policlinico Gemelli in Rome we had the opportunity to setup and clinically validate a profile for the detection of relevant mutations over BRCA1 and BRCA2 genes, that are known to be involved in breast and ovarian cancer.
Obtained results allowed us to achieve CE-IVD certification for analysis of BRCA datasets with Amplicon Suite, but most importantly demonstrated that by setting up a proper analysis protocol it is possible to apply NGS techniques to clinical routine while getting the same accuracy provided by traditional methods.

SmartVir is a user friendly tool for virus genotyping and drug resistance detection. It was developed in collaboration with Roche Italian sequencing team with particular focus on the creation of a profile for analysis of samples from HCV infected patients.
In just few clicks, the user with no specific bioinformatics skills can accurately detect the specific strain (or strains in case of mixed infections) of HCV that is affecting the patient and probe viral genome in order to determine the presence of mutations that are know to be associated with drug resistances.
An automatic integration procedure compares detected mutations with dedicated external authoritative databases (currently Geno2Pheno [2] and Lontok [3]) in order to collect the most up to date informations about them.
The final output is an intuitive report about relevant mutations detected for each sample with informations about frequencies and possible drug resistances that clinicians can use to determine the most suitable treatment option.
All obtained results were validated in collaboration with Roche Italian sequencing team. The immediate future perspective for SmartVir is

the introduction of support for analysis of HIV.

Finally we also presented STAble, a new assembler for RNA-seq data that is built around a completely original approach where the whole reads are used to drive the assembly process instead of considering only smaller k-mers.
It achieves a sensitivity comparable to existing tools while significantly decreasing the number of false positive reconstructions, that in a real denovo environment can lead to production of unreliable results, since there is almost no way to detect them.
Moreover it is designed to be parallelizable, allowing to break down the onerous assembly task into smaller subsets that can be processed in parallel even on common desktop computers.
The possibility to apply RNA-seq denovo assembly techniques to enhance metatranscriptome studies was also explored by properly tuning STAble: preliminary results on simulated datasets are confirming the viability of this approach.
The immediate future perspective is its application to real metatranscriptome studies in collaboration with the local hospital.
The long-term goal is the integration of this feature into **MicrobAT** [4], an user-friendly tool for profiling microbial populations developed my research group that at present is capable of microbiota identification.
The idea is to transform it into an integrated suite for complete metagenomics studies: microbiota profiling, metatranscriptome assembly, annotation of new transcripts, mapping of transcripts to metabolic pathways and application of *flux balance analysis* [5] to better understand the balances that are established within the bacterial community.

5.1 Acknowledgements

Amplicon Suite, *SmartVir* and *MicrobAT* were developed in collaboration with *SmartSeq s.r.l.*, spin-off of the University of Piemonte Orientale. They are hold and marketed by *SmartSeq s.r.l.*.

Bibliography

[1] den Dunnen JT, Dalgleish R, Maglott DR, Hart RK, Greenblatt MS, McGowan-Jordan J, Roux AF, Smith T, Antonarakis SE, Taschner PE. *HGVS Recommendations for the Description of Sequence Variants: 2016 Update.* Hum Mutat. 2016 Jun; 37(6): 564-9.

[2] Prabhav Kalaghatgi, Anna Maria Sikorski, Elena Knops, Daniel Rupp, Saleta Sierra, Eva Heger, Maria Neumann-Fraune, Bastian Beggel, Andreas Walker, Jörg Timm, Hauke Walter, Martin Obermeier, Rolf Kaiser, Ralf Bartenschlager and Thomas Lengauer. *Geno2pheno[HCV] – A Web-based Interpretation System to Support Hepatitis C Treatment Decisions in the Era of Direct-Acting Antiviral Agents.* PLoS One 2016; 11(5): e0155869.

[3] Lontok E, Harrington P, Howe A, Kieffer T, Lennerstrand J, Lenz O, McPhee F, Mo H, Parkin N, Pilot-Matias T, Miller V. *Hepatitis C virus drug resistance-associated substitutions: State of the art summary.* Hepatology. 2015 Nov; 62(5):1623-32.

[4] Palladini A & Boatti L, Saggese I, Paroni Sterbini F, Masucci L, Sanguinetti M, Mignone F *MicrobAT - a user-friendly package for profiling microbial populations.* Human Gut Microbiome and Diseases; 2015 Jun 25-26; Milan – IT.

[5] Mark Hanemaaijer, Wilfred F. M. Röling, Brett G. Olivier, Ruchir A. Khandelwal, Bas Teusink, and Frank J. Bruggeman. *Systems modeling approaches for microbial community studies: from metagenomics to inference of the community structure.* Front Microbiol. 2015; 6: 213.

Chapter 6

List of Publications

6.1 Journals and book chapters

- Minucci A, Scambia G, Santonocito C, Concolino P, Canu G, Mignone F, **Saggese I**, Guarino D, Costella A, Molinario R, De Bonis M, Ferrandina G, Petrillo M, Scaglione GL, Capoluongo E.
  Clinical impact on ovarian cancer patients of massive parallel sequencing for BRCA mutation detection: the experience at Gemelli hospital and a literature review.
  Expert Rev Mol Diagn. 2015;15(10):1383-403.

- Boria I, Boatti L, **Saggese I**, Mignone F.
  NGS-Trex: an automatic analysis workflow for RNA-Seq data.
  Methods Mol Biol. 2015;1269:243-56.

6.2 Participation in conferences

- **Igor Saggese**, Giovanni Manzini and Flavio Mignone.
  STAble: a novel approach to denovo assembly of RNA-seq data.
  13th International Meeting CIBB 2016; 2016 Sept 1-3; Stirling - UK.

- Palladini A & Boatti L, **Saggese I**, Paroni Sterbini F, Masucci L, Sanguinetti M, Mignone F.

MicrobAT - a user-friendly package for profiling microbial populations.
Human Gut Microbiome and Diseases; 2015 Jun 25-26; Milan – IT.

- Marina Martello PhD, Barbara Santacroce, Angela Flores Dico, Enrica Borsi PhD, Torsten Haferlach, Flavio Mignone, **Igor Saggese**, Elena Zamagni MD, Paola Tacchetti MD, Lucia Pantani MD, Annamaria Brioli MD, Beatrice Anna Zannetti MD, Serena Rocchi MD, Katia Mancuso MD, Nicoletta Testoni BSc, Giulia Marzocchi PhD, Gaia Ameli BSc, Annalisa Pezzi, Michele Cavo MD, Giovanni Martinelli MD and Carolina Terragna PhD.
  A Long Tail of Sub-Clonal TP53 Mutations Emerged By Ultra-Deep Sequencing of Newly Diagnosed Multiple Myeloma (MM).
  56th ASH Annual Meeting and Exposition; 2014 Dec 6-9; San Francisco - CA.

- Scaglione G L, Guarino D, Concolino P, Santonocito C, Mignone F, **Saggese I**, Costella A, Minucci A, Capoluongo E.
  Performance evaluation of Next Generation Sequencing (NGS) for BRCA1/2 testing by a novel bioinformatics analysis tool (Amplicon Suite).
  IFCC WorldLab Istanbul 2014; 2014 Jun 22-26; Istanbul - TR.