UPPSALA
UNIVERSITET

UPTEC X 14 014
Examensarbete 30 hp
November 2014

Assessment of variant load in an idiopathic autoinflammatory index patient

Jessika Nordin

UPPSALA UNIVERSITET

Degree Project in Bioinformatics

Masters Programme in Molecular Biotechnology Engineering,
Uppsala University School of Engineering

UPTEC X 14 014	**Date of issue 2014-11**

Author

Jessika Nordin

Title (English)

Assessment of variant load in an idiopathic autoinflammatory index patient

Title (Swedish)

Abstract

An index patient with an idiopathic autoinflammatory diseased was sequenced for over ~1900 immunological genes, and their regulatory elements, in a Targeted Sequence Capture Library. This data was used for creating a bioinformatics pipeline for all projects that use the same library. The pipeline was build from the GATK best practises framework and goes from raw sequence data to a list with ranked variants.

To receive a list of interesting variants, the index patient was compared to his immediate family and a cohort of Swedish controls. This was done since it is probable that the disease causing variants in the index patient is private to him (the family do not have the variant). The controls were used to be sure that the variants are not common in the Swedish population.

Keywords

Autoinflammatory disease, bioinformatics pipeline, GATK, index patient, variation load

Supervisors

Jennifer Meadows
Uppsala University, IMBIM

Scientific reviewer

Alvaro Martinez Barrio
Uppsala University, IMBIM

Project name	Sponsors
Language **English**	Security
ISSN 1401-2138	Classification
Supplementary bibliographical information	Pages **39**

Biology Education Centre Biomedical Center Husargatan 3, Uppsala
Box 592, S-751 24 Uppsala Tel +46 (0)18 4710000 Fax +46 (0)18 471 4687

Assessment of variant load in an idiopathic autoinflammatory index patient

Jessika Nordin

Populärvetenskaplig sammanfattning

Autoinflammatoriska sjukdomar är ett relativt nytt begrepp inom medicin som dök upp för ca 15 år sedan. Autoinflammatoriska sjukdomar uppstår på grund av fel i det medfödda immunförsvaret (till skillnad mot autoimmuna sjukdomar som uppstår på grund av fel i det specifika immunförsvaret). Det medfödda immunförsvaret är det som reagerar först mot främmande föremål i kroppen. Spannet av autoinflammatoriska sjukdomar är brett och kan vara orsakade av en eller flera gener. Flera av sjukdomarna delar orsak och symptom vilket gör dem svåra att diagnosera och behandla.

För att kunna ta reda på mer om de olika autoinflammatoriska sjukdomarna har man tagit fram ett riktat sekvensfångande bibliotek som innehåller 1900 gener och deras reglerande element. Detta bibliotek har använts på 115 kontroller och en familj där den ena sonen har en oidentifierad autoinflammatorisk sjukdom. Pojken har en blandning av olika symptom som är unika i hans fall och ingen medicin hjälper helt mot symptomen. En bioinformatisk pipeline sattes upp för att smidigt analysera sekvensbiblioteket. Med hjälp av den övriga familjen och kontrollerna har vi tagit fram en lista med tänkbara varianter som kan vara orsaken till pojkens sjukdom. Denna lista ska utvärderas och kan förhoppningsvis hjälpa till att förbättra pojkens vård.

Examensarbete 30 hp

Civilingenjörsprogrammet i Bioinformatik

Uppsala universitet, november 2014

Table of Contents

Glossary

AID	autoinflammatory disease
AN	number of alleles with data
BAM	binary SAM file
BWA	Burrows-Wheeler Aligner
CLL	chronic lymphocytic leukaemia
CNV	copy-number variations
DNA	deoxyribonucleic acid
EMR1	epidermal growth factor *(EGF)-like module containing, mucin-like, hormone Receptor-like 1*
Eosinophils & neutrophils	two kinds of white blood cells
FAM	individual information file (family ID, individual ID, paternal ID, maternal ID, sex, phenotype)
FS	Fisher strand which is a phred-scaled p-value to detect strand bias
GATK	Genome Analysis Toolkit
GB	gigabyte
Hg 19	latest build of the human genome
HS	hybrid selection
Idiopathic	unknown origin to the disease
IL	Interleukin
IMBIM	Department of Medical Biochemistry and Microbiology
MB	megabyte
MEFV	Mediterranean fever
MNP	multi nucleotide polymorphism

MQ	root mean square of the mapping quality of the reads across all samples
MQRankSum	an approximation of the Mann-Whitney rank sum test for mapping qualities
NCBI	National Center for Biotechnology Information
NGS	next generation sequencing
NLRP3	NACHT, LRR and PYD domains-containing protein 3
NRD	non-reference discrepancy
NRS	non-reference sensitivity
OGC	overall genotype concordance
QD	quality by depth calculated by the variant confidence divided by the unfiltered depth of non-reference samples
readPosRankSum	an approximation of Mann-Whitney rank sum test for the distance from the end of the read for reads with alternate alleles
SAM	sequence alignment/map file
SNP	single nucleotide polymorphism
SpA	spondylitis arthritis
Target	the part of the genome that the library is made to capture
TNFRSF1A	tumor necrosis factor receptor superfamily
Uppmax	Uppsala Multidisciplinary Center for Advanced Computational Science
UTR	untranslated region
Variant	where the sample differs from the reference
VCF	variant calling format file
VQSR	variant quality score recalibrator

Introduction

Autoinflammatory disease (AID) is a quite new field in medicine and was discovered only 10-15 years ago[1]. AIDs are caused by errors in the innate immune system (the immune system we are born with). The innate immune system is the first to react to danger signals inside or outside of the cells, before the acquired immune response takes over. These diseases are widely spread in how they appear and what causes them, some of them are monogenic and others are multifactorial. But the thing these rare diseases all have in common is that they cause episodes of inflammation in patients, without any sign of autoantibodies or antigen-specific T-cells[2]. Many of the different diseases also share symptoms, which makes diagnosis and treatment challenging[3].

Not much is known about the genetics of AID. At present, the EUROFEVER register (http://www.printo.it/eurofever/) contains a list of 18 genes with known variants associated with AID. The remaining 80% of patients have no known genetic cause of disease, hindering the potential application of medicines to target pathways of AID[3].

In collaboration with the comparative genomics arm of the Department of Medical Biochemistry and Microbiology (IMBIM) at Uppsala University, we have studied a number of different heritable immunological disorders, and have for this purpose developed the custom NimbleGen Targeted Sequence Capture Library that was used in this thesis. The objective of this array was to cover the coding and regulatory regions of approximately 1900 genes that are involved in immune responses. For each disorder, paired-end Illumina Next Generation Sequencing was used to assay this ~32 MB (or ~1% of the human genome) to a depth of 10x for many hundreds of individuals within specific case and control groups. In practise, one sample (500 ng of DNA) from the targeted Sequence Capture Library gives approximately 2-6 GB data in the form of two fastq files after sequencing. Since the sample was pooled with seven other samples, each lane produced 14-20 GB raw data. For instance, in this thesis the analysis was based on 122 samples that gave 354 GB raw data. A bioinformatics pipeline was implemented to handle this huge amount of data.

If it is challenging to treat the autoinflammatory diseases that are known, it is even harder to treat an idiopathic autoinflammatory disease. In this paper we tried to find the cause of an idiopathic disease using the above methodology to generate data from an index patient, six members of his immediate family and a cohort of control genomes. The first part of this process involved the building of a bioinformatics pipeline that could be used across projects to analyse the result of the custom library. The second stage was the implementation of this procedure to generate a list of potential disease causing variants for further evaluation.

Methods

Sample Resources

The male index patient had been diagnosed with an idiopathic autoinflammatory disease and the rest of the family were healthy (except the grandfather on the mother's side that had chronic lymphocytic leukaemia (CLL)). The combination of symptoms was unique for the index patient and clinicians had not found a medicine that worked completely; the medicines only lessen the symptoms. The index patient had hypersplenism (over activity and enlargement of the spleen), fever, skin eruption, problems with his joints and had had aseptic meningitis (meningitis without any sign of bacterial involvement). No one in the family apart from him showed any signs of autoinflammatory disease (figure 1).

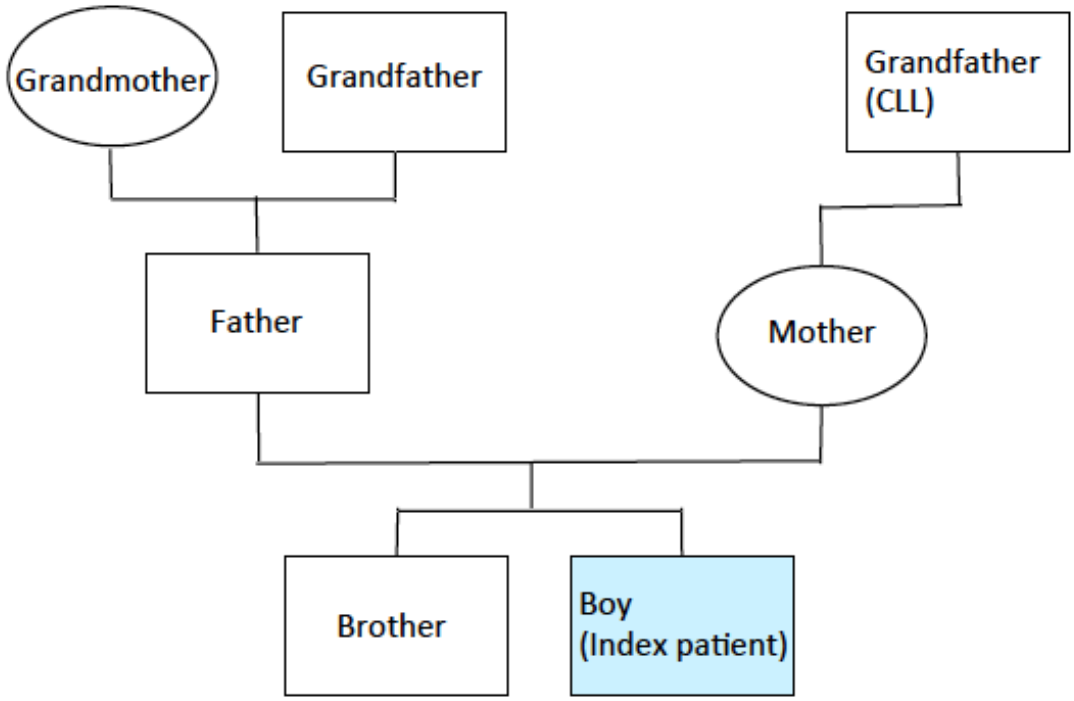


Figure 1 | Family pedigree
Figure 1 shows the relationship between the index patient and the close members of his family. All were sequenced as part of this project.

The index patient did not respond to an interleukin 1 (IL-1) blockade, but had for a long time been given Prednisolone (a medical preparation with corticosteroid that is used for allergies and rheumatic trouble), which had a positive effect. Prednisolone decreases inflammation and makes the immune response less active by reducing cytokine gene expression and promoting apoptosis of eosinophils[4,5]. In July 2013 he was prescribed an IL-6 blockade. Currently, his dosage of Prednisolone is being reduced. He was also given Colchicine. Colchicine is a substance from a flower called autumn crocus, meadow saffron or naked lady. It is an anti-inflammatory medicine that prevents neutrophil motility and activity[6]. This medicine is only made on demand.

Blood samples were taken 2013-08-27 from the family (figure 1). DNA from these samples were extracted and libraries prepared (at IMBIM, Uppsala

University) and sent to the Sequencing Centre (Science for Life Laboratory). How the samples were prepared will not be discussed in this thesis. The raw data from the Sequencing Centre came back 2013-12-16, and the amount of data from each sample is shown in table 1.

Table 1 | Amount of data per sample
This table shows how much data came back from the Sequencing Centre for each sample.

Sample	Amount of data (GB)
Boy (index patient)	6.0
Brother	3.2
Mother	4.4
Father	3.8
Grandmother (father's side)	3.8
Grandfather (father's side)	6.6
Grandfather (mother's side)	2.8
Control group (115 samples)*	4.2

*For the controls the mean value for all 115 samples is used.

As the index patient was idiopathic, it was likely that a specific combination of disease causing variants were private to him in comparison to his immediate family. However, to place potentially interesting variants in context, we also considered a control group. This group was taken from the control cohort in the ankylosing spondylitis (SpA) project running at IMBIM, and consisted of 115 samples. These controls were considered to be of Swedish ancestry (both the sequenced individual and their parents were born in Sweden) and were from the same geographical area as the index patient (south of Sweden). This control group was important as a variant that was identified in the index patient, but that does not exist in the general population (e.g. 1000 genomes), may in fact be common in the Swedish population, and so this variant's interest in a disease context would be down weighed. All samples used for analysis in this work were obtained following approved ethical protocols (Dnr M177-07).

Sequenced samples in this project were received as raw Illumina HiSeq pair-end 100 bp fastq reads.

Development of the Sequencing Pipeline

The targeted Sequencing Capture Library that was used for the index patient and his family will also be used to generate samples in complementary immunological projects. In order to facilitate the exchange of data between these experiments, it was essential that raw data from each individual project was processed in the same way, to reduce analyses biases which may skew important values such as coverage and allele frequency. A bioinformatics pipeline was implemented to be able to analyse the data. The family was used as a test project for constructing the pipeline that will be used for similar future projects.

The foundation for the pipeline comes from Genome Analysis Toolkit's (GATK) best practices[7]. This pipeline requires a lot of computational power and a subset of software modules. For this purpose, Uppsala Multidisciplinary Center for

Advanced Computational Science (Uppmax, https://www.uppmax.uu.se), was used. Uppmax is a resource of high-performance computers with a number of software tools already installed.

Data pre-processing

When the data came back from the Sequencing Centre it needed to be pre-processed before any kind of analysis, like searching for variants, could take place. This was done with help of SAMtools (samtools.sourceforge.net), Picard (picard.sourceforge.net) and GATK (www.broadinstitute.org/gatk), as shown in figure 2.

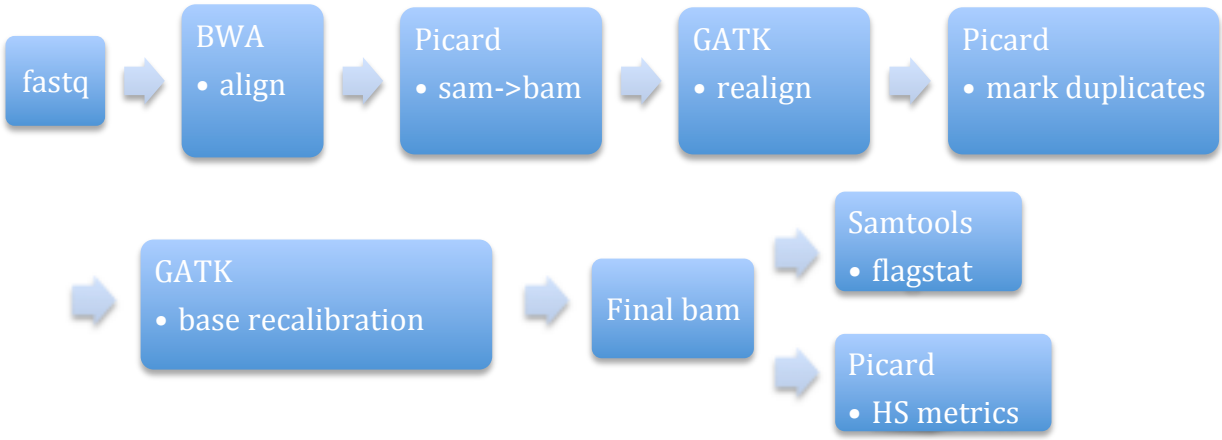


Figure 2 | Flowchart of pre-processing
The first script is called finalbam.sh and does all the data pre-processing. It takes the raw data in fastq format and uses BWA, Picard, GATK and SAMtools to align, realign, mark duplicates, do a base recalibration and generates two quality files, flagstat and HS metrics and the cleaned data in a finalbam.bam.

The Burrows-Wheeler Aligner (BWA) was used to align the fastq files to the human genome (in this case version hg19 augmented with chr6_cox_haplotype2 and chr19_random)[8]. BWA 0.7.4 was used for this, rather than the newer 0.7.5a, which had a bug that ended the aligning without warning in some cases (it was not known what activated this bug). BWA 0.7.4 did not show any signs of having the same bug. This process created a SAM file. SAM files are big plain text files (for example one sample had a 21 GB SAM file) that are hard to handle because there is no way to access subsets of data quickly[9]. Picard was used to convert the SAM file into a BAM file. A BAM file is a SAM file in binary, which take less space (the same sample as before has a 5 GB BAM file) and it can be indexed, which makes it easy to randomly access data quickly[9].

Both the physical sequencing and the data handling give minor errors and the BAM file contained many of these artefacts. In an attempt to reduce the amount of errors we cleaned the BAM file. Because of the close collaboration between the IMBIM group and Broad Institute, GATK (that is developed by Broad Institute) best practices were used as the frame of the pipeline. The GATK developer provided a data bundle that contained most of the files necessary to be able to do the best practices, all from reference genomes to Mills indels (a gold-standard set of indels that were validated separately), which helped much in our downstream process[10]. However, our reference genome was not taken from GATKs data

bundle. GATK 2.7.2 was used because it was the fastest version of GATK at Uppmax at the time of project commencement. The first step after creating the first BAM file was to go through the file and find intervals that had indels, this interval file was then used to go through the BAM and realign at these intervals. The next step to mark duplicates was done by Picard. Biological duplicates exist, e.g. copy-number variations (CNV), but duplicates can also be created during the experiment. Duplicates can be created by PCR amplification during library construction, or by reading the same fragment twice during sequencing. Marked duplicates were removed from downstream analyses, the main reason for this was to remove the effects of PCR amplification bias. Then, the data was recalibrated against dbSNP common SNP list (version 137) so the program could see what a SNP should look like, and give those bases a recalibration score[11]. The last thing before the data was finally ready was to get some quality information for a number of different statistics. These statistics were then used to decide if a sample would be going further downstream in the analysis pipeline, or if the data from a sample was too poor. For example, in this analysis, a mean coverage of at least 10x was necessary for good results with other tools in the pipeline.

The quality of the final BAM file was judged using a variety of statistics. SAMtools was used to get for example the amount of duplicates, mapped reads, properly paired reads and singletons with the command flagstats. Picard was used to get hybrid selection metrics (HS metrics), which give for example number of reads on and off target, mean coverage and how many bases that had 10x, 20x and 30x coverage. These quality tests gave more data that was taken into consideration further downstream in the analysis. Now the final BAM was ready to be analysed after cleaning.

Variant discovery

When the final BAM was finished it could proceed to the variant calling that was done in two steps; call variants (which create a VCF file[12]) and filter the called variants (figure 3). GATK has two different routes for variant calling and for filtering. Two walkers were available for variant calling (walkers is the different methods inside of GATK), unifiedGenotyper and haplotypeCaller, and for filtering, hard filters and variantRecalibrator, that uses Variant Quality Score Recalibration (VQSR)[13].

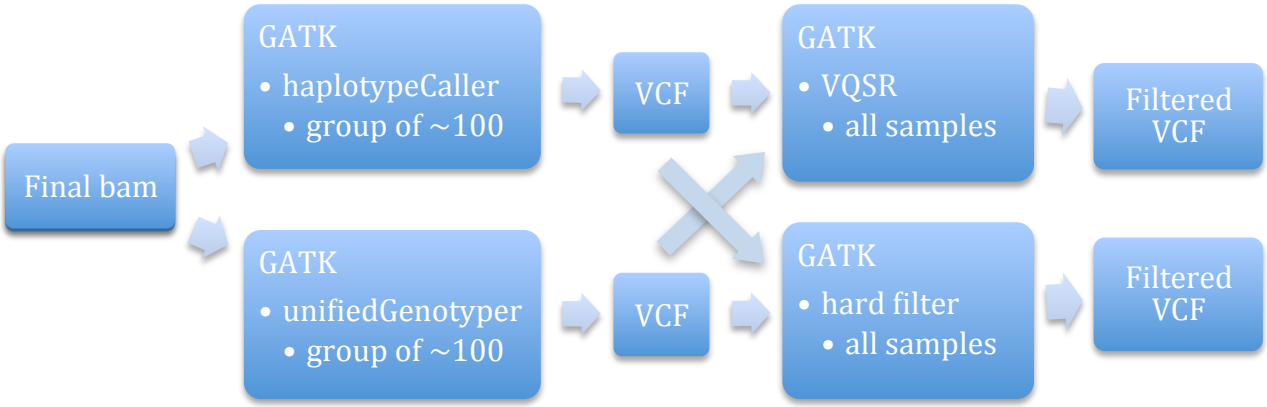


Figure 3 | Flowchart of the variant calling
Variant calling is made on the final BAM with two scripts. haplotypeCaller.sh or unifiedGenotyper.sh takes a group of sample BAMs (~100) and runs the variant calling per chromosome in parallel and gives a VCF as an output file. VariantRecalibration.sh and hardFilter.sh are both used to take all samples and use various metrics to filter variants based on assigned quality thresholds.

HaplotypeCaller is recommended for diploid organisms. However, it requires more time and has a sample limit of ~100/run. GATK recommends haplotypeCaller because it is a more advanced tool. Instead of simply calling variants, it takes an interval region around each variant and applies a *de novo* realignment to verify that the result alignment is the optimal. Consequently, this process is time consuming because it requires a lot more computational power.

UnifiedGenotyper is better at handling different numbers of chromosome copies, from one to several. UnifiedGenotyper is not that good at finding indels, and picks up false variants that are actually alignment errors due to BWA. BWA is good for fast aligning, if you are ready to let the quality of the alignment slip a little. With indel realign in the data pre-process step and haplotypeCaller, these mistakes from BWA are dealt with, but it can be noticeable when running unifiedGenotyper.

Another problem with the variant calling was if variants should be called individually or as an individual within a group of samples. It was decided to do the variant calling in a group after a test with the index patient and his three closest family members (brother, father and mother). Variants were called with haplotypeCaller, both individually and as a group for the family. This small test was made to investigate if the number of alleles that were covered (the AN number in a VCF files information field) differed depending on how the variants were called. The theory was that if you called them individually, only positions with variants would be exported to the VCF, positions equal to the reference would not. But if the samples were called in a group it would be enough if one sample had a variant, to export that position for all samples, variant or not. (In the new patch of GATK, GATK 3.0, haplotypeCaller was able to call variants on individual samples but haplotypeCaller was not there yet in the version that was used in this study).

HaplotypeCaller vs. unifiedGenotyper and hard filter vs. VQSR

To decide what kind of method and which walker to use, an experiment with the two walkers and the filtering methods was designed. A group a 32 samples was selected, since GATK suggested that around 30 samples for exome sequencing is needed for VQSR to work properly (the 32 samples were picked because they were easily accessible at that point). Variant calling on these 32 samples were performed in eight different ways as shown in figure 4. The variants were called individually for each sample, by both unifiedGenotyper and haplotypeCaller, and all 32 together in a group (but not merged) by unifiedGenotyper and haplotypeCaller. We then post-processed all these different out-files by both VQSR and hard filter.

Method evaluation was completed using ten different intersections following variant annotation on snpEff (version 3.4) that gave a comparison on how different/similar the different methods were.

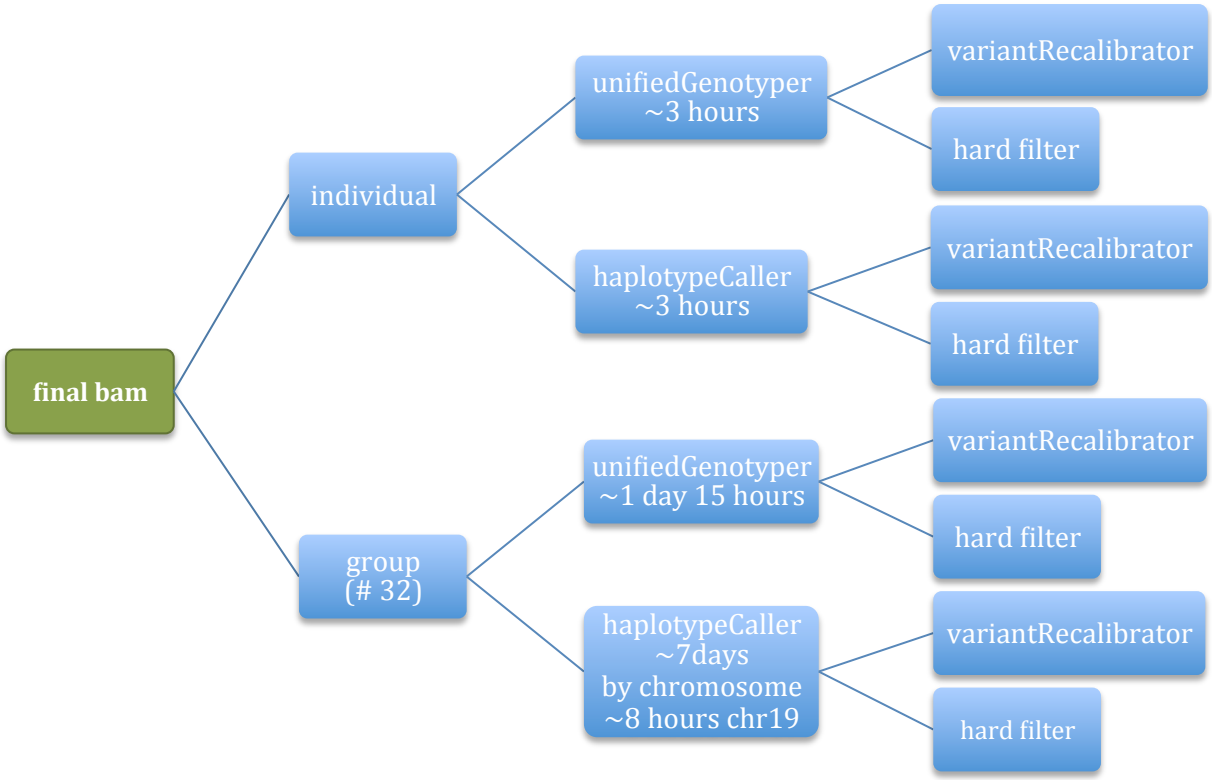


Figure 4 | HaplotypeCaller vs. unifiedGenotyper and hard filter vs. VQSR
To compare the different walkers to each other a test with 32 samples was made. All 32 samples started as an analysis ready BAM file (final BAM) and then they were handled individually or in a group. The eight analyses methods were tested using these 32 samples and then compared.

Variant calling

More than 100 control samples were available for analysis in the project, however ~100 samples was the input limit for haplotypeCaller (because of time issues). For this reason, groups of ~100 controls plus the index patient were used to ensure all individuals had variant, or reference positions, exported to complement all his variants. To be able to run the haplotypeCaller in a reasonable timeframe, the samples were divided into chromosomes and each

chromosome was run in parallel. This decreased the time required to a maximum of 3 days (for chromosome 3 and 116 samples) compared to the more than seven days to process whole genomes for 32 samples (data not shown. Note, whole genomes for 116 samples were not attempted).

After this step, there were VCF files with the variants in the index patient and the equivalent information for the other samples. But some variants were called even if they were not that good. To only get the best candidates for real variants, the called variants needed to be filtered. Once again there were two ways to do this. One was the old and well tested hard filtering, which is dependent on your information on what a good variant is. The other was VQSR that uses machine learning to decide what a good variant looks like[13].

Hard filtering uses a number of variables when doing its filtering. Users can easily change variables used after what fits the purpose of the experiment. In this experiment the default values were used. For filtering SNPs, QD, FS, MQ, MQRankSum and readPosRankSum were used. QD is quality by depth calculated by the variant confidence divided by the unfiltered depth of non-reference samples and was set to 2.0. FS is Fisher strand, which is a phred-scaled p-value to detect strand bias and was set to 60.0. MQ is root mean square of the mapping quality, of the reads across all samples, and was set to 40.0. MQRankSum is an approximation of the Mann-Whitney rank sum test for mapping qualities, and was set to 12.5. ReadPosRankSum is an approximation of the Mann-Whitney rank sum test for the distance from the end of the read for reads with alternate alleles and was set to 8.0. While filtering indels, only QD, FS and ReadPosRankSum were used; where QD =2.0, FS =200.0 and ReadPosRankSum = -20.0[13].

The VQSR uses a machine learning algorithm to filter variants. VQSR is given datasets with real SNPs (1000 genomes project phase 1, SNP database 137, HapMap 3.3 and Omni 2.5) and indels (Mills indels) that are already validated and have different priorities related to the confidence that they are true variants. For example, 1000 genomes is a set of high-confidence SNP sites, which the machine learning program will consider to have both true variants and false positives and have a rank of 10. dbSNP have not been validated to a high degree of confidence and have therefore the lowest rank of only 2. The variant calling dataset was produced from the family and controls. VQSR scores all variants depending on where they end up in the model. Then the scores are ordered and depending on which interval the score is in, it can be filtered or not. There are different intervals that can be used depending on what the individual project's need for variant sensitivity and specificity. For our experiment, a sensitivity of 99.0 was used. It was easy to access the variants from the higher sensitivity because they got the grade of sensitivity for which they belonged. This was done for SNPs and indels separately because there are different models for SNPs and indels. After this process the variants were ready for examination.

Genotyping

Both the index patient and 11 of the 115 control individuals had been genotyped to some extent before.

The index patient was genotyped for three different genomic regions at the Institution for Clinical and Experimental Medicine at Linköping University. They used Sanger sequencing to look at rs35829419 in *NLRP3*, exons 1, 2, 3, 5 and 10 in *MEFV* and exons 2, 3 and 4 in *TNFRSF1A*. The DNA sequences were then compared against the hg19 reference sequence to find mutations. We used the Linköping sequences as controls when examining if the variant calling methodology developed here was working properly.

A subset of the controls were genotyped using the Affymetrix Genome-Wide Human SNP Array 6.0, again at Linköping University. This affy-array contained more than 906,600 SNPs. However, not all of these variants were covered by the Targeted Sequence Capture Library and so not all were available for comparison to those produced by the variant calling.

Preliminary analysis

Once variants were called it was time to try to find possible disease causing variants. GATK recommends snpEff 3.4 (snpeff.sourceforge.net*)*, which is a variant annotation tool, and it was used to annotate which effect a variant can have. SnpEff looks at the variants to see where they are, in a gene, intron, 3' or 5' UTR and so on. It also predicts different kinds of effects the variant can have (frame shift, start lost, stop gained and so on and so forth) and these effects are classed into how deleterious they can be (high, moderate, low and modifier, which often means affecting non-coding regions) [14].

After the annotation of the variants effects, snpSift 3.4 was used for a number of analyses: caseControl, private, heterozygote and homozygote[15]. SnpSift's caseControl was given the filtered variants with their effects and a FAM file of the samples. (A FAM file contains family id, sample id, mother, father, sex and affected/unaffected, but all information does not have to be there for it to work.) CaseControl looks at each variant and tells how many alleles are homozygote, heterozygote and variants in total for cases and for controls separately. Since running caseControl on big files is time consuming, the files were split into chromosomes, and caseControl ran them separately, and in parallel, to decrease the time to get to a result.

The caseControl output made it easy to filter out variants that were private for cases (only the cases that had alleles that had changed for that position), homozygote for cases (where cases were homozygote and no controls were homozygote, these variants could be recessive and possibly disease causing) or heterozygote for cases (where cases were heterozygote and no controls were heterozygote). There are examples, like the heterozygote 667X mutant in the murine model for an endocrine disease Familial neurohypophyseal diabetes insipidus (FNDI), where a heterozygote variant was more deleterious than to be homozygote alternate allele, so we can not exclude them as candidates. This analysis was repeated to evaluate i) index patient as case and his mother, father

and brother as control, ii) index patient as case and the rest of the family as controls, iii) index patient as case and the control group as controls. These different analyses produced three files containing private, homozygote and heterozygote variants. A private variant does not necessarily mean that it is of interest even though it would fit the pattern of idiopathic disease. A private variant can be deleterious but it can also have no effect at all. Two different lists were created, one with all variants from the three files which have high effect and one with those that have moderate effect (figure 5).

Figure 5 | Flowchart of the preliminary analysis
In this step we have three scripts. SnpEff.sh is used on the VQSR filtered variants and annotates each variant. caseControl.sh takes the annotated file and does snpSifts caseControl on each chromosome in parallel. Then snpSift.sh is used to do filtering and get variants that are private, heterozygote or homozygote in the cases. Variants with predicted effect of high and moderate are used to make variant lists.

The variants that appeared in all three analyses were the most interesting and had priority for further information gathering. Also, genes that contained many variants were interesting. To be able to see if a variant seemed interesting, the different analyses were compared. Therefore, the information about the family's genotype and the control group's allele frequency was gathered for each variant. Also, the depth for the variant in the family and the number of reads called for reference/alternate genotype in the index patient was taken into consideration. The evaluation of variants started by using the UCSC genome browser (http://genome-euro.ucsc.edu) to see if the variant was known and its allele frequency in different populations[16]. To find out more about what was known for the variant, Online Mendelian Inheritance in Man® (http://www.omim.org/) was used. Either the gene name or the snp number was used as a search keyword[17]. For variants that caused changes in the amino acid sequence, the National Center for Biotechnology Information (NCBI, www.ncbi.nlm.nih.gov) was used to find which protein position a variant had. This was to see if the changed amino acid seemed to interfere in important functions. For genes with several variants it was interesting to see if there were different haplotypes present. This analysis was done with PHASE[18].

Results

Data pre-processing

Quality information was gathered from the first step of the pipeline. SAMtools flagstat gave mapping statistics such as number of reads, number of duplicates, number of mapped, number of properly paired and singletons. Picard's HS metrics gave information about how many target bases that had coverage of 2x, 10x, 20x and 30x. HS metrics also gave information about the mean coverage for the sample. All this information can be found in table 2 for the close family and the controls (mean values).

The mean coverage over all samples was higher than 18x, which was more than the required 10x coverage to continue the analyses. All samples had more than 78% of the target bases with at least 10x coverage. The boy (index patient) had most raw data while his brother was the sample with least raw data. The number of duplicates was around 60% in the family while it was half in the controls. The opposite was true for bases that mapped off bait, 16% of the controls bases were mapped off bait while it was 8% for the family.

Table 2 | Quality information from flagstat and HS metrics
The most important information from flagstat and HS metrics is compiled in this table.

Quality information	Boy	Brother	Father	Mother	Controls*
Raw size (Gb)	6.0	3.2	3.8	4.4	4.2
Reads in total	59,152,084	32,888,602	37,608,408	43,423,036	42,176,334.12
Duplicates	35,810,667	19,931,422	22,405,347	26,968,033	13,088,902.66
Duplicates %	60.54%	60.60%	59.58%	62.11%	31.18%
Mapped	58,202,195	32,405,662	37,060,522	42,777,628	41,218,898.48
Mapped %	98.39%	98.53%	98.54%	98.51%	97.72%
Properly paired	57,770,664	32,174,438	36,794,298	42,462,818	40,909,455.23
Properly paired %	97.66%	97.83%	97.84%	97.79%	96.98%
Singletons	324,829	167,346	190,310	227,426	202,705.38
Singletons %	0.55%	0.51%	0.51%	0.52%	0.48%
Target bases with at least 2X	96.46%	95.43%	95.87%	96.14%	96.69%
Target bases with at least 10X	90.18%	78.18%	82.49%	85.74%	90.61%
Target bases with at least 20X	75.13%	42.51%	52.21%	60.06%	77.15%
Target bases with at least 30X	54.13%	16.04%	24.57%	32.21%	59.23%
Target bases with < 2X	1.88%	2.27%	2.08%	2.04%	1.70%
Mean coverage	33.21	18.66	21.64	24.31	37.55
Bases that mapped off bait	8.07%	7.98%	8.09%	8.06%	16.72%
Bases on bait/bases passed the vendor filter	91.93%	92.02%	91.91%	91.94%	83.28%

* The control value is given as the mean of the 115 samples.

Variant discovery

Variant calling, individual or in group

After running the closest family members in haplotypeCaller the number of different AN values were counted. AN numbers are always even because it counts alleles and alleles are in a chromosome pair. AN=2 means that the position was covered on each chromosome in one sample. AN=4 means in two samples and so on. This was done to see if information could be lost about positions when variants were called individually. The working hypothesis was that if a position was like the reference it would not be recorded in the VCF file. Then, for a variant that would be private for the index patient it would be impossible to say if the others were covered as they would have a reference allele at that position, or not be covered at all. That would create a problem because it was important for the analysis to know if the others were reference or if there was just no data. Table 3 shows a clear difference between how many positions were recorded in each sample, for example a decrease from 18% to 5% of AN=2 calls.

Table 3 | HaplotypeCaller test with family samples
The closest family members were used to find out the difference of AN (number of alleles covered in a position) if the samples were going through the haplotypeCaller individually or as a group.

haplotypeCaller	Individually	(%)	Group	(%)
Total variants	136,012		160,495	
AN=2	24,001	18	8,100	5
AN=4	36,185	27	7,467	5
AN=6	31,291	23	10,606	7
AN=8	44,535	33	134,322	84

HaplotypeCaller vs. unifiedGenotyper and hard filter vs. VQSR

In a comparison of the two walkers, haplotypeCaller gave more variants when samples were called individually than unifiedGenotyper, and the opposite was true when the samples were called in a group, for this test with 32 randomly chosen samples. One thing to keep in mind is that haplotypeCaller found indels in the data set while unifiedGenotyper did not call any indels. When the different walker outputs were intersected for individual and group calling the result showed that around one and two thirds, respectively, of the variants overlap (shown in table 4). Of the variants that overlap, around 400,000 variants were only found in one sample (AN=2) for the individuall calling, whilst this dropped 1,000-fold in the group calling, to 409.

Table 4 | HaplotypeCaller vs. unifiedGenotyper
This table shows the number of variants found when using different variant calling methods. UnifiedGenotyper and haplotypeCaller were run on 32 samples both individually and as a group. Intersect is how many variants that overlap of the variants above in the table, and AN=2 intersect tells how many variants that were covered in only one sample.

# variants	Individual	Group
UnifiedGenotyper (UG)	6,233,106	6,839,297
HaplotypeCaller (HC)	6,392,655	6,632,747
Intersect of UG and HC	2,369,163	4,642,346
AN=2 intersect	375,667	409

The four different variant calling files were filtered with both hard filtering and VQSR. UnifiedGenotyper gave 2-4 million variants that passed the filtering, while haplotypeCaller gave 5-6 million and the filtering methods differed by 100,000-300,000 variants, where hard filtering gave more. Hard filtering gave more variants with max AN for all combinations. VQSR however kept more AN=2 than hard filtering when the variant calling was done in group mode, while hard filtering gave more AN=2 when the calling was done individually (data shown in appendix, table S1). When intersecting the filtered group variant calling files, it showed that almost every variant with AN=2 from hard filtering also was kept by VQSR. Overall, it was the other way around, both number of passed variants, AN=32 and AN=64 from VQSR were almost completely covered by hard filtering. SnpEff was used to get more information about the filtered variants files. It showed that haplotypeCaller found multi-nucleotide polymorphism (MNP), indels and deletions when unifiedGenotyper didnot find any of those. It was a clear increase (between two to seven times as much) in the number of MNP, indels and deletions found, when the variants were called in a group. Also more variants over all were called.

Genotyping

Both our genotyping efforts and Linköping University's results showed that the index patient was wildtype for all regions except the rs35829419 in *NLRP3* where he was homozygote for an alternative allele.

For the 906,600 SNPs from the affy-array, only 84,500 SNPs were covered by the Targeted Sequence Capture Library, theoretically. Even if the targeted Sequence Capture Library was supposed to cover all those positions, in practise this might not be true due to poor coverage. In contrast, other parts of the genome could be covered which were not planned because off-target capture. It could also be the case that the SNPs were all homozygote to the reference and therefore not exported to the VCF during variant calling and VQSR. When changing the affy-data to VCF format we lost some SNPs on the way, both when lifting the affy-data from hg18 to hg19 and when removing SNPs that were not covered in the affy data. This left us with 867,350 SNPs from the affy-array.

When comparing the variants found in the 11 controls with the SNPs from the affy-array it was possible to do that in several ways. In this thesis three different methods were tested. Method 1 compared the affy-array VCF directly without

transforming the reference build or SNP call. Method 2 used the SNP data generated by using haplotypeCaller to export variants only for the list found in the affy array (here the affy-data was transformed so that the same reference positions corresponded between both files). Method 3 was a standard haplotype-Caller on the controls and the affy-data, which overlapped with ours. These approaches gave very different results, especially in regards to how many variants were available for the concordance test (table 5). Method 2 had 413 variants available while method 3 had 6,002. Both showed that for the alleles available for comparison, there was a greater than 98.7% concordance.

As well as site level concordance, the above tests also provided levels of genotype concordance, i.e: Non-reference sensitivity (NRS) tells us the proportion of polymorphic SNPs from the affy-data that were also polymorphic in our data. Non-reference discrepancy (NRD) tells us the proportion of sites that were not concordant when excluding the concordant reference sites. Overall genotype concordance (OGC), is the proportion of concordant genotypes we have, in comparison to all genotypes. Table 6 shows that the overall genotype concordance was high with method 3 (OGC = 0.80), although, on the individual level, OGC ranged from 0.690-0.968 (see appendix, table S2).

Table 7 clearly shows the effect of variants being unavailable and where concordance is lost. For example, with method 3, when variants were called heterozygous in the affy-data, our data recorded a majority of homozygous variant alleles (22.1%) as opposed to the expected heterozygotes (11.2%).

Table 5 | Site concordance: site-level summary statistics
A summary of the three methods for genotype concordance between our data and the affy-array. This shows that all analyses gave different results.

	Method 1	Method 2	Method 3
Variants available (our data/affy data)	764,648 / 2,235,583	413 / 2,235,586	6,022 / 2,235,586
Alleles match	99,568	408	6,009
Alleles do not match*	142,001	2	0
Our data only	523,071	3	13
Affy data only	1,994,006	2,235,176	2,229,577

* Alleles do not match: counts of calls at the same location with different alleles, such as the array-data set calling a 'G' alternate allele, and our data set calling a 'T' alternate allele.

Table 6 | Genotype concordance: summary statistics
A summary of the three methods for genotype concordance between our data and the affy-array.

	Method 1	Method 2	Method 3
Non-reference sensitivity	0,059	0,314	0,262
Non-reference discrepancy	0,714	0,000	0,002
Overall genotype concordance	0,441	0,770	0,800

Table 7 | Genotype concordance: proportions of genotypes called in relation to the affy data

This table summarise what call our data had when compared to the affy-array for method 2 and 3. This shows for which calls the concordance was low or high. Many genotypes could not be compared because the data did not exist (i.e "No_call" and "Unavailable" in "Our genotype").

		Method 2		Method 3	
Affy genotype	**Our genotype**	Count	Proportion	Count	Proportion
Het*	**Het**	138	0.133	776	0.112
	Hom ref	67	0.065	539	0.078
	Hom var	143	0.138	1523	0.221
	Mixed	0	0	0	0
	No call	684	0.66	4036	0.584
	Unavailable	5	0.005	33	0.005
Hom ref*	**Het**	16	0.012	156	0.007
	Hom ref	285	0.216	2944	0.128
	Hom var	5	0.004	88	0.004
	Mixed	0	0	0	0
	No call	1004	0.762	19681	0.858
	Unavailable	8	0.006	80	0.003
Hom var*	**Het**	8	0.004	149	0.004
	Hom ref	8	0.004	36	0.001
	Hom var	402	0.183	6257	0.173
	Mixed	0	0	0	0
	No call	1761	0.801	29751	0.821
	Unavailable	20	0.009	30	0.001

* Het: heterozygote; Hom ref: homozygote for reference allele; Het var: Heterozygote for variant allele.

Preliminary analysis

Variant list

From the caseControl analyses, made using the idiopathic boy's alleles, a list of 231 variants was composed. 122 variants were found in the comparison with his closest family, 86 variants were found with his whole family and 77 variants were found with the controls. 138 variants were found by more than one analysis method (where seven variants were found by all three) and three were classed both as having high and moderate effect. 19 of these variants were predicted to have high effect and 212 were predicted to have moderate effect (the high effect list can be found in the appendix, table S3). Of these 231 variants, 89 were called as private in at least one of the three analyses. 137 variants were called as homozygote and 94 as heterozygote. Seven of the variants with moderate effect were found by all three analyses.

One gene that stood out in the analysis was epidermal growth factor *(EGF)-like module containing, mucin-like, hormone Receptor-like 1* (*EMR1)*. Most genes (172/191) had only one variant and a few (19/191) had between two and five. *EMR1* however, had ten variants, one had both high and moderate effects

predicted (chr19: 6897464). All of these ten variants were known SNPs. The index patient was homozygote for all these variants while his close family was heterozygote for all positions. However, the grandfather on his mother's side who developed CLL, was also homozygote for all these variants. When looking at the controls (all 115 controls were covered for each variant), no individual was homozygote for all these variants. The allele frequency for the alternate allele is shown in table 8. Five of the variants were in, or in close proximity to, a calcium binding motif. Variant rs330877 was seldom (~3.3% of the time) homozygote and it is located in between two amino acids that are used for making disulfide bonds in the protein. When running this set of variants with PHASE, we found that the index patient's affected haplotype was not uncommon (10.4%). But having that haplotype as a pair was only seen in the index patient and the grandfather (1.6%).

Table 8 | EMR1 variants

For the gene *EMR1,* ten variants were found. This table presents where the variants are, their SNP reference number, the reference allele and the alternate allele. The index patient and grandfather on the mother's side have the same alleles and the index patient's father, mother and brother have the same allele and are included in the table in the column close family. The allele frequency is for the alternative allele in the control group. All 115 controls were covered for all variants.

Chr	Position	SNP	Ref	Alt	Boy and grandfather (mother's side)	Close family	Allele freq. alt from controls
chr19	6896483	rs330877	G	A	A/A	G/A	0.252
	6897464	rs330880	C	G	G/G	C/G	0.317
	6901891	rs897738	G	A	A/A	G/A	0.265
	6903920	rs443658	A	G	G/G	A/G	0.265
	6904137	rs370094	C	T	T/T	C/T	0.265
	6913707	rs466876	C	T	T/T	C/T	0.196
	6913811	rs457857	A	G	G/G	A/G	0.713
	6919624	rs373533	A	C	C/C	A/C	0.735
	6919753	rs461645	A	G	G/G	A/G	0.730
	6926378	rs2228539	T	C	C/C	T/C	0.257

Discussion

Data pre-processing

A great deal of effort was spent on stream-lining the amount of time and processing power required to perform the data pre-processing. From the initial pipeline build, which required ~17 hours for a sample of 6 GB, with the usage of updated, debugged software, we were able to shorten this to ~5 hours. The initial pipeline used GATK 2.3.6, Picard 1.69 and BWA 0.7.5a. As mentioned earlier, BWA 0.7.5a was exchanged with 0.7.4 because of a bug. GATK 2.3.6 was exchanged with 2.7.2 and Picard 1.69 was exchanged with 1.92, because 2.7.2 and 1.92 are newer and would reduce the required time. Time estimation graphs for both versions can be found in the appendix, figure S1. The pipeline works well in its current form and no private scripting is needed. However, this pipeline was made to be as user friendly as possible through implementation of variables and lists. One example where variables were used was for the reference, which could be altered by changing one row of code instead of every row that uses the reference. The variables were listed in the beginning of the scripts and were therefore easy for users to change. For the scripts where it was possible, lists were used to save time for the user. These scripts were adapted to run for each sample in a list instead of having to be changed by hand for every sample. For example, a list was used as an input set of fastqs for finalbam.sh to generate final BAMs.

Even though the scripts were tested and error searches were done, the input of new data can result in yet unseen bugs. These do not always terminate the run; it continues until the end and there is no captured sign that something negative has happened. It would be beneficial to put in some security points along the way in the script so that it crashes as soon as an error occurs. This would be especially good in finalbam.sh, which always runs all steps and gives a long error output file that makes it hard to find the causative error. To have a security point after each step would decrease the computational time and it would be possible to find the error without reading the whole output file. There is also room for improvements when it comes to simplifying the script so that any person easily can make changes and use it.

The pre-processing script was within this thesis successfully applied to 122 samples. The quality files generated at this point in the process, as part of the finalbam.sh, showed that each sample had at least 18x coverage, far exceeding the 10x required to proceed to variant discovery.

Variant discovery

Variant calling, individual or in group

The results (table 3) clearly showed that performing the variant calling in a group is preferable. The main reason for this was that data for reference positions were still saved if another sample had a variant in that position. If the variant calling was performed individually only, positions where that individual was variant would be exported. There would be no way of knowing if missing data was due to that the sample was a homozygote reference or if it was not

covered. Then, when trying to see if a variant was important or not you would be missing data crucial for that decision. This information is especially important when trying to find a variant that can be disease causing, like in this thesis. It also makes it easier to calculate the allele frequency since the number of samples that are covered for a position is available from the beginning. The frequency will be calculated for all samples and not only for the samples that have a variant, which it would be if the samples were called individually. To perform the variant calling in group also gives additional support to call variants, which give more variants when samples are called in group than if they are called individually. 136,012 variants were called when the family was called individually, while calling variants for the family as a group gave 160,495 variants.

Because it was the index patient's variants that were of most interest to this thesis, the index patient was used to call for variants in the controls. That ensured that the data for all the index patient's variants were saved in the controls too. The allele frequency for a variant in the control group was good to have when evaluating the variants in the variant list. If a variant had a high frequency in the control group it was probably not disease causing, however a variant can have a high frequency in a population because of many heterozygotes and still be deleterious when homozygote. All factors have to play in for the decision and the allele frequency is one of the important elements.

HaplotypeCaller vs. unifiedGenotyper and hard filter vs. VQSR

HaplotypeCaller proved to be the best walker to use in the current context. One reason for this was the de-novo assembly used by haplotypeCaller in regions where it found variation. This reduced the number of variants called because of bad aligning. Another reason was that haplotypeCaller found more variants that passed the filtering. UnifiedGenotyper found 6,839,297 variants, whereof 5,164,458 passed the filtering, compared to haplotypeCaller that found 6,632,747 variants, whereof 5,389,646 passed. HaplotypeCaller was also much better at getting variants other than just SNPs, such as indels and MNP.

Determining which method between hard filtering and VQSR that was the best to use was not easy. Both methods had good qualities. The decision about the most convenient walker depends on the user and on the project. If the user has a good idea of how different quality values should be regulated, then hard filtering is good. With hard filtering it is clear how and why a variant is retained. It is a strict method and there are no grey areas. VQSR is a machine learning method and it makes its own decisions, which can be good if the project design allows for using it. VQSR builds a Gaussian mixture model, unique for the datasets, that are used to create the model and your dataset is fitted to it. This makes it easy to use because you do not have to choose parameters, only give the program good datasets to build the model on. However, the model can only be used on the dataset that were used to create it, which means that all samples have to be filtered at the same time to be sure that they are treated the same way. There was no big time difference between these two methods so that factor was not part of the decision making. At the end we decided that the best thing to do was to run both methods if possible and then start with analysing the variants that passed both. For example, when looking at the experiment with the 32 samples

there were 6,629,925 variants found with haplotypeCaller. Hard filter and VQSR gave 5,609,072 and 5,389,646 variants, respectively, that passed the filtering. When looking at variants that both had in common, there were still 5,292,374 variants left to be analysed. In this thesis, only VQSR variants were analysed in depth, even though hard filtering was also performed. Future analyses will use the intersect of these two methodologies.

Genotyping

In order to give confidence, genotyping calls generated through our pipeline were examined by the concordance between two data sets, the index patient and Sanger data, and 11 controls and affy-data. These two comparisons between both technologies were done to strengthen the credibility of our results and show that Illumina NGS data at our depth of coverage was good enough as to emit true genotype calls. We observed the same result for the three regions in the index patient. For the 11 controls it was harder to evaluate the concordance.

The difficulties in generating genotype concordance for the 11 controls will be discussed here. The affy array data was called on hg18 while our data was called on hg19. This can be easily fixed, but some data will be lost when changing it from hg18 to hg19. When the affy array data was analysed, both the positive strand and the negative strand were used when deciding reference, while for our reference alleles, data was only from the positive strand. If a negative strand called variant from the affy array was compared with our data it would not be in concordance, even if they actually were, because they were each other's reverse (for example A in affy array and T in our data). This "strandness" can be fixed, but some data can be lost in this step too. The affy array data came as MAP and PED files, while the program used for check concordance only uses VCFs. When converting MAP and PED to VCF there were some problems too. The scripts that were easily accessible used the major allele as the reference which was not always true (www.pypedia.com/index.php/bioinformatics_format_convert). So here, a script needs to be created to be used to change all SNPs which were given the wrong reference. One more problem was that our data only saved positions where there was a variant. So all positions that were homozygote references were not available for the genotype concordance, and 61.59% of the affy array data were homozygote references which caused us to lose much data that could have been used.

As shown in table 7, there were many homozygote genotypes called in our data where the affy data called a heterozygote genotype. This can have different explanations; it could be that our data had too low coverage at that position and so the heterozygotes were not observed, or it could be a reflection of how the affy array data was filtered. At this moment it is not known how the affy array data was filtered, and clearly this was important when looking closer at the genotype concordance. This also indicated that maybe more filtering should be done on our data to get a credible result. For example, a depth filter and a genotype quality filter would help with identifying the best variants, and hopefully help with receiving a good genotype concordance.

GATK has evolved to work with larger population data sets (1000 and more samples) and that advance is going to simplify this process in the future (not shown in this thesis). The newest GATK haplotypeCaller, 3.2.2, makes calls for all positions, so the positions that are not variant will now be saved and used in the genotype concordance. It should then be easier to get the positions that overlap between the both data sets and it should give a better indication on the genotype concordance.

Preliminary analysis

Variant list

HaplotypeCaller and VQSR were used to make the variant list. There were several interesting variants in the list, which all had to be evaluated. The variants investigated first were the ones found by all three caseControl analyses, followed by those densely distributed within genes. It was good to have a three step study design, where we zoom out from the index patient. It started with the closest family, this analysis gave some variants which the next two steps would not have caught. The gene *EMR1* would not have been found if it was not for this. Because the index patient and the grandfather on the mother's side looked the same, no variants for *EMR1* would have made it to the variant list when looking on the whole family. When it came to the controls, there were always some controls that were homozygote for the same variant and would have hidden this very interesting gene. However, when looking at the different phased haplotypes for these ten variants, there were only 10% of the population (122 samples) that had that haplotype and only the index patient and his grandfather were homozygote for the haplotypic pair. This could indicate that this combination might be deleterious.

In this thesis we have more closely examined a particularly interesting set of variants found in the index patient. Firstly, he is homozygous for a mutation in *NLRP3* (pQ705K), a gain of function mutation that has been shown to lead to an increased release of the pro-inflammatory cytokines IL-1b and IL-18[19]. Secondly, he is homozygous for ten mutations within the gene, *EMR1* (Table 8). Unlike its mouse homologue, F4/80, the human seven-transmembrane *EMR1* is absent on mononuclear myloid cells and it is instead predominantly produced by eosinophils[20]. Knockout mice show that absence of *EMR* leads to a lack of efficient regulatory T-cell development[21]. But the role of *EMR1* in human is unknown, although it could be highly specific given its limited cellular distribution.

EMR1 was of particular interest because it had variants with both high and moderate effect according to snpEff and a heavier variant load than other genes. *EMR1* had ten variants that made the variant list, compared with the other genes that had one to five variants. We know that the grandfather on the mother's side shares the index patient's variants, however, this gene could still be important because the grandfather has chronic lymphocytic leukaemia (CLL) and he is heterozygote for the mutation pQ705K in *NLRP3*. CLL has two subgroups and can affect B-cells or T-cells. As we know, *EMR1* is located on the eosinophils, which play a role in Th2-inducing agents, and an error in *EMR1* could initiate Th2 immunity[22].

Of the ten variants in *EMR1*, three (rs897738, rs466876, rs2228539) were considered more likely to be deleterious. The first, an Asp to Asn amino acid charge change, likely results in the loss of the second EGF domain to bind a required calcium molecule. Given previous 3D crystallographic studies on a closely related family member, *EMR2*[23], we suspect that this would weaken the stabilisation of the extracellular domain of the receptor and affect its ability to bind ligands. The next mutation causes a hydrophilic (Thr) to hydrophobic (Met) amino acid change in the extracellular G-protein-coupled receptors (GPCR) autoproteolysis-inducing (GAIN) domain. The GAIN domain is essential for adhesion-GPCR function and disruption of this may affect efficient ligand binding[24]. The third homozygote variant is a reverse of the second; this time a hydrophobic (Met) amino acid is altered to be hydrophilic (Thr) and may disrupt an N-linked glycosylation site, although a Thr at this position is observed in mouse[25].

Whilst the potential action each of the mutations may have on the function of this eosinophil receptor is unknown, we do know that these cells are essential for inducing Th2 immunity through both the promotion of Th2 differentiation and the recruitment of effector Th2 cells to required sites within tissues[25]. In fact, eosinophils are a key source of the cytokines (e.g. IL-4, IL-13 and IL-6) required for Th2 induction. These cytokines are stored in intracellular granules, allowing for rapid release without the need for de novo synthesis[22]. Eosinophils themselves respond to cytokine stimulus, and a constitutively active NLRP3 inflammasome driven by the mutation (pQ705K) may actually provide the trigger for this downstream inflammatory event.

A clinical examination of blood counts and inflammatory markers was conducted on the index patient following the partial withdrawal of corticosteroid treatment, Prednisolone (unpublished results). There it was shown that he was on, or below, the lower limit of eosinophil and lymphocyte reference values (0.02×10^9/L and 1.5×10^9/L, respectively). In addition, his circulating IL-18 concentration was more than five times (2700 pg/mL) above the maximum reference value during treatment. Each of these measurements are in keeping with the working hypothesis of overactive inflammasome and attenuated eosinophil action. It would be of interest to also measure the cytokines released from the eosinophil granules to see if the EMR1 mutations prevent cyokine release or eosinophil development.

While we have two particularly interesting genes, there is still more to find out about the index patient. The index patient has a complex disease and it seems likely that there are both autoinflammatory and autoimmune components at work. There is still more work that could be done with EMR1 (experiments to see if the variants actually cause problems), also more controls could be added and the rest of the variant list could be evaluated more carefully.

Future work

There are a number of things that can be done before the pipeline is finished and before the index patient's variants have been thoroughly analysed.

Pipeline

As I mentioned before, there are still changes that can be made to make the scripts more user friendly. The first thing is to make all the scripts as good as possible and make some kind of documentation on how to use them. It might be possible to create an actual pipeline that calls all the scripts so that the user only has to handle one script. That would be the best solution.

Preliminary analysis

To get the best result possible it would be good to use another, but similar, tool for the effect prediction. ANNOVAR (www.openbioinformatics.org/annovar) could be used to be able to compare between tools and work towards the best result. ANNOVAR have more information to use when annotating. It does not predict effect but gives a lot of different scores.

It would also have been nice to be able to perform a gene ontology analysis on the variants observed, to check if any overrepresented category could be found. The idea is to compare both the control group and the family of the affected individual. The control group would function as a reference on what would be normal. This could reveal an affected pathway, with several variants causing disturbance.

Acknowledgements

First of all I want to express my deepest gratitude to my fantastic supervisor Jennifer Meadows that did everything (and more) to help and guide me through my thesis.

I also want to thank my scientific reviewer Alvaro Martinez Barrio who took time to look at my reports, and who gave good critic and asked good questions.

I want to thank everybody involved in making this project possible.
To Kerstin Lindblad-Toh and Cecilia Johansson for support and meetings to discuss my progress.
To Gerli Pielberg that designed the Targeted Sequence Capture Library used.
To Peter Söderkvist at Linköping University who supplied the control samples and the genotyping data mentioned in the report.
To Stefan Berg at University of Gothenburg that supplied the family's samples and phenotyping.
To A. Iris Mathioudaki, Fabiana Farias and Johanna Dahlqvist for asking questions on how to use the pipeline which was a help to me during the development.
And to A. Iris Mathioudaki, again, for all the labwork and support during this thesis.

I would also like to thank Uppmax and Science for Life Laboratory's Sequencing Centre at Uppsala University for their services. And also the Uppmax support, which is quick on answering questions and requests.

I am particularly grateful for the assistance given by Mårten Larsson on the subject of understanding what the different variants in EMR1 could result in.

I want to give many thanks to Matilda Åslin that agreed to be my opponent. She has been a great support during the whole project (from how to write "~"on a Mac, to how to write the acknowledgements).

And the last thanks go to everybody that I have been in contact with during this thesis, especially my family and friends.

References

1 McDermott M.F, Aksentijevich I, Galon. J, McDermott E.M, Ogunkolade B.W, Centola M, Mansfield E, Gadina M, Karenko L, Pettersson T, McCarthy J, Frucht D.M, Aringer M, Torosyan Y, Teppo A.M, Wilson M, Karaarslan H.M, Wan Y, Todd I, Wood G, Schlimgen R, Kumarajeewa T.R, Cooper S.M, Vella J.P, Amos C.I, Mulley J, Quane K.A, Molloy M.G, Ranki A, Powell R.J, Hitman G.A, O'Shea J.J, Kastner D.L. 1999. Germline mutations in the extracellular domains of the 55kDa TNF receptor (TNF-R1) define a family of dominantly inherited autoinflammatory syndromes. *Cell*. 97:133-44.

2 McDermott M.F, Aksentijevich I. 2002. The autoinflammatory syndromes. *Curr Opin Allergy Clin Immunol.* 2:511-6.

3 Toplak N, Frenkel J, Ozen S, Lachmann H.J, Woo P, Koné-Paut I, De Benedetti F, Neven B, Hofer M, Doiezalova P, Kümmerle-Deschner J, Touitou I, Hentgen V, Simon A, Girschick H, Rose C, Wouters C, Vesely R, Arostegui J, Stojanov S, Ozgodan H, Martini A, Ruperto N, Gattomo M, Paediatric Rheumatology International Trails Organisation, Eurotraps and Eurofever Projects. 2012. An international registry on autoinflammatory diseases: the Eurofever experience. *Ann Rheum Dis*. 71:1177-82.

4 Fernandes S, McKay G. 2013. Prednisolone. *Practical diabetes*. 30:251–252.

5 Robinson D, Hamid Q, Ying S, Bentley A, Assoufi B, Durham S, Kay B. 1993. Prednisolone treatment in asthma is associated with modulation of bronchoalveolar lavage cell interleukin-4, interleukin-5, and interferon-γ cytokine gene expression. *Am Rev Respir Dis*. 148:401-406.

6 Stack J, Ryan J, McCarthy G. 2013. Colchicine: New insights to an old drug. *Am J Ther*. E-pub.

7 McKenna A, 2010. The Genome Analysis Toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res*. 20:1297–1303.

8 Li H, Durbin R. 2009. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*. 25:1754–1760.

9 Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R, 1000 Genome Project Data Processing Subgroup. 2009. The Sequence Alignment/Map format and SAMtools. *Bioinformatics*. 25:2078–2079.

10 Van der Auwera G.A, Carneiro M.O, Hartl C, Poplin R, del Angel G, Levy-Moonshine A, Jordan T, Shakir K, Roazen D, Thibault J, Banks E, Garimella K.V, Altshuler D, Gabriel S, DePristo M.A. 2013. From FastQ Data to High-Confidence Variant Calls: The Genome Analysis Toolkit Best Practices Pipeline. *Protoc Bioinform*. 43:11.10.1-11.10.33.

11 Sherry S, Ward M-H, Kholodov M, Baker J, Phan L, Smigielski E, Sirotkin K. 2001. dbSNP: The NCBI database of genetic variation. *Nucleic Acids Res*. 29:308–311.

12 Danecek P, Auton A, Abecasis G, Albers C.A, Banks W, DePristo M.A, Handsaker R.E, Lunter G, Marth G.T, Sherry S.T, McVean G, Durbin R, 1000 Genomes Project Analysis Group. 2011. The variant call format and VCFtools. *Bioinformatics.* 27:2156-2158.

13 DePristo M.A, Banks E, Poplin. R.E, Garimella K.V, Maguire J.R, Hartl C, Philippakis A.A, del Angel G, Rivas M.A, Hanna M, McKenna A, Fennell T.J, Kernytsky A.M, Sivachenko A.Y, Cibulshis K, Gabriel S.B, Altshuler D, Daly M.J. 2011. A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nat Genet*. 43:491–498.

14 Cingolani P. 2012. A program for annotating and predicting the effects of single nucleotide polymorphisms, SnpEff: SNPs in the genome of Drosophila melanogaster strain w1118; iso-2; iso-3. *Fly (Austin).* 6:80-92.

15 Cingolani P. 2012. Using Drosophila melanogaster as a model for genotoxic chemical mutational studies with a new program, SnpSift. *Frontiers in Genetics*. 3.

16 Kent W.J, Sugnet C.W, Furey T.S, Roskin K.M, Pringle T.H, Zahler A.M, Haussler D. 2002. The human genome browser at UCSC. *Genome Res.*12:996-1006.

17 Online Mendelian Inheritance in Man, OMIM®. McKusick-Nathans Institute of Genetic Medicine, Johns Hopkins University (Baltimore, MD), 2014-03-21. World Wide Web URL: http://omim.org/.

18 Stephens M, Smith N, Donnelly P. 2001. A new statistical method for haplotype reconstruction from population data. *American Journal of Human Genetics.* 68:978-989.

19 Verma D, Särndahl E, Andersson H, Eriksson P, Fredrikson M, Jönsson J.I, Lerm M, Söderkvist P. 2012. The Q705K polymorphism in NLRP3 is a gain-of-function alteration leading to excessive interleukin-1β and IL-18 production. *PLoS One*. 7:e34977.

20 Hamann J, Koning N, Pouwels W, Ulfman LH, van Eijk M, Stacey M, Lin H.H, Gordon S, Kwakkenbos M.J. 2007. EMR1, the human homolog of F4/80, is an eosinophil-specific receptor. *Eur J Immunol*. 37:2797-2802.

21 Mouse genome informatics. Jackson Laboratory. 2014-05-07. World Wide Web URL: http://www.informatics.jax.org/marker/MGI:106912.

22 Spencer L.A, Weller P.F. 2010. Eosinophils and Th2 immunity: contemporary insights. *Immunol Cell Biol.* 88:250-256.

23 Abbott R.J, Spendlove I, Roversi P, Fitzgibbon H, Knott. V, Teriete P, McDonnell J.M, Handford P.A, Lea S.M. 2007. Structural and functional characterization of a novel T cell receptor co-regulatory protein complex, CD97-CD55. *J Biol Chem*. 282:22023-22032.

24 Prömel S, Langenhan T, Araç D. 2013. Matching structure with function: the GAIN domain of Adhesion-GPCR and PKD1-like proteins. *Trends Pharmacol Sci*. 34:470-478.

25 McKnight A.J, Gordon S. 1998. The EGF-TM7 family: unusual structures at the leukocyte surface. *J Leukoc Biol*. 63:271-280. Review

Appendix

Table S1 | Hard filter vs. VQSR
The table shows the number of variants that were found with unifiedGenotyper (UG) and haplotypeCaller (HC) when 32 samples were processed a) individually or b) in group. It shows how many variants that past with the different filtering methods, hard filter and VQSR. Different AN were compared for how many variants there were from the beginning and how many were left after filtering, to be able to see how the different filtration methods worked.

a)	Individual UG hard filter	Individual UG VQSR	Individual HC hard filter	Individual HC VQSR
# variants	6,233,106	6,233,106	6,392,323	6,392,323
# of passed	3,708,444	3,215,165	2,761,016	2,417,792
AN=2	3,068,823	3,068,823	3,180,058	3,180,084
AN=2 passed	898,932	768,855	594,037	475,346
AN=32	10,497	10,497	9,500	9,507
AN=32 passed	10,497	7,106	9,484	7,155
AN=64	15,507	15,507	12,306	12,306
AN=64 passed	15,507	9,235	10,998	10,051

b)	Group UG hard filter	Group UG VQSR	Group HC hard filter	Group HC VQSR
# variants	6,839,297	6,839,297	6,629,925	6,629,925
# of passed	5,286,851	5,164,458	5,609,072	5,389,646
AN=2	20,168	20,168	3,670	3,670
AN=2 passed	440	772	2,153	2,819
AN=32	420,217	420,217	454,391	454,391
AN=32 passed	362,060	358,457	384,807	375,675
AN=64	283,004	283,004	271,974	271,974
AN=64 passed	206,936	127,677	235,967	187,117

Table S2 | Genotype concordance: summary statistics per sample of NRS, NRD, and OGC
Three tables with per sample summary from each method for genotype concordance. A) Method 1, b) Method 2 and c) Method 3. In this table it is possible to see if each sample behaves the same with the different methods.

a)	ALL	SpA-CO51_tag72	SpA-CO51_tag74	SpA_CO56_tag4	SpA_CO62_tag83	SpA_CO65_tag73	SpA_CO65_tag85	SpA_CO65_tag90	SpA_CO66_tag66	SpA_CO69_tag68	SpA_CO69_tag95	SpA_CO71_tag13
Non-Reference Sensitivity	0,059	0,062	0,058	0,053	0,059	0,058	0,063	0,064	0,064	0,059	0,059	0,059
Non-Reference Discrepancy	0,714	0,590	0,658	0,623	0,568	0,613	0,518	0,510	0,536	0,794	0,756	0,773
Overall_Genotype_Concordance	0,441	0,603	0,531	0,579	0,627	0,561	0,667	0,679	0,656	0,347	0,352	0,344

b)	ALL	SpA-CO51_tag72	SpA-CO51_tag74	SpA_CO56_tag4	SpA_CO62_tag83	SpA_CO65_tag73	SpA_CO65_tag85	SpA_CO65_tag90	SpA_CO66_tag66	SpA_CO69_tag68	SpA_CO69_tag95	SpA_CO71_tag13
Non-Reference Sensitivity	0,314	0,222	0,300	0,148	0,111	0,560	0,250	0,167	0,217	0,325	0,220	0,430
Non-Reference Discrepancy	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
Overall_Genotype_Concordance	0,770	0,872	0,793	0,917	0,953	0,600	0,825	0,875	0,848	0,751	0,841	0,637

c)	ALL	SpA-CO51_tag72	SpA-CO51_tag74	SpA_CO56_tag4	SpA_CO62_tag83	SpA_CO65_tag73	SpA_CO65_tag85	SpA_CO65_tag90	SpA_CO66_tag66	SpA_CO69_tag68	SpA_CO69_tag95	SpA_CO71_tag13
Non-Reference Sensitivity	0,262	0,102	0,253	0,062	0,086	0,509	0,299	0,093	0,087	0,361	0,056	0,356
Non-Reference Discrepancy	0,002	0,001	0,001	0,001	0,001	0,001	0,001	0,001	0,001	0,003	0,002	0,003
Overall_Genotype_Concordance	0,800	0,949	0,845	0,968	0,957	0,709	0,819	0,952	0,951	0,690	0,959	0,690

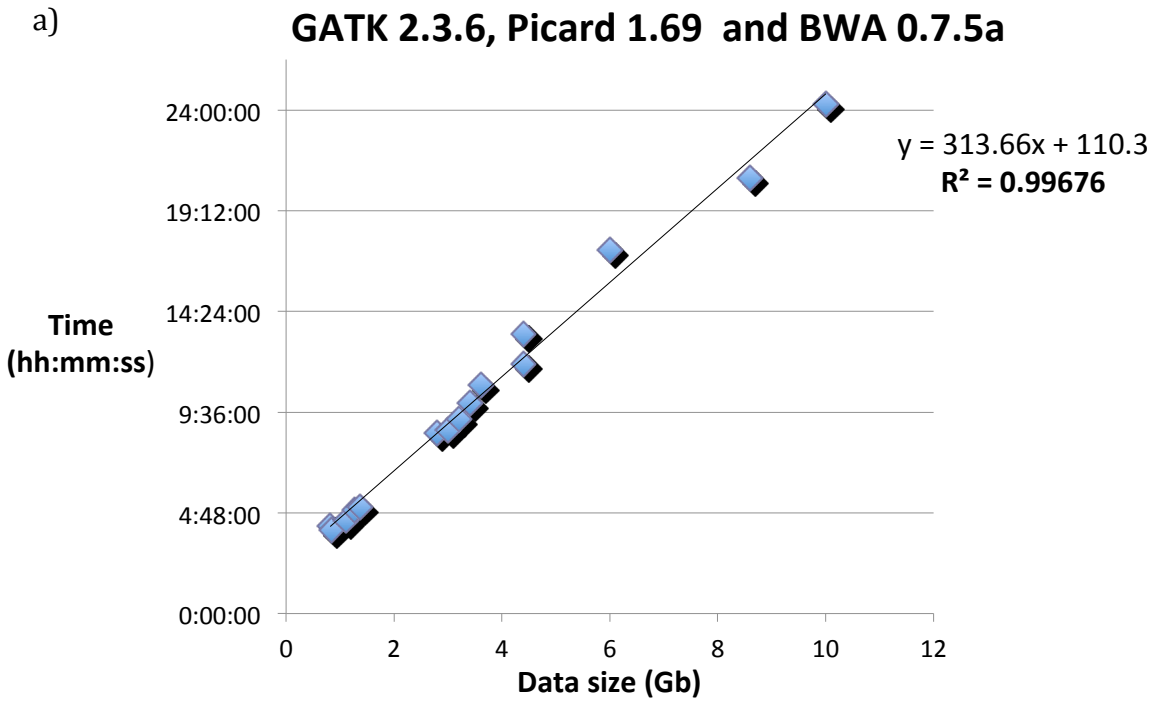
Table S3 | Variant list

To be able to show the variant list it is split in two with the 8 first columns being the same. The first part is with the index patient and his close family and the other part is with the index patient and the rest of the family. This variant list contains the 19 variants that were predicted to have high effect.

Chr	Pos	From	Type	SNP	Ref	Alt	Boy	Brother	Mother	Father	Allele freq. alt	Gene
chr1	17085995	close/ family	priv/het		G	GC	G/GC	G/G	G/G	G/G	0,079	MST1P9
chr1	109650648	family	priv/het		A	T	A/T	-	-	-	0,04	C1orf194
chr1	207758220	close	priv/het		TG	T	TG/T	TG/TG	TG/TG	TG/TG	-	CR1
chr2	69659126	close/ family	hom	rs4453725	A	T	T/T	A/T	A/T	A/T	0,422	NFU1
chr2	135675908	close	hom	rs1348793	T	C	C/C	T/T	T/C	T/T	0,239	CCNT2
chr3	46414696	control	priv/het		T	A	T/A	T/T	T/A 17/33	T/T	0	CCR5
chr5	149563573	close/ family	priv/hom	rs73796347	A	G	G/G	-	-	-	0,17	AC005895.4
chr10	99376152	close	hom	rs3814556	A	G	G/G	A/G	A/G	A/G	0,252	MORN4
chr14	21458320	close/ family	Hom	rs147038155	CCCCG	C	C/C	CCCCG/C	CCCCG/C	CCCCG/C	0,326	METTL17
chr14	64855040	control	priv/het		CT	C	CT/C	CT/CT	CT/CT	CT/C	0	MTHFD1
chr15	49867232	control	priv/het		C	T	C/T	C/C	C/C	C/T	0	C15orf33
chr17	41063408	control	priv/het		C	T	C/T	C/T	C/C	C/T	0	G6PC
chr17	45567593	close/ family	hom		ATC	A	A/A	ATC/A	ATC/A	ATC/A	0,5	AC040934.1
chr19	6897464	close	hom	rs330880	C	G	G/G	C/G	C/G	C/G	0,317	EMR1
chr19	10577801	control	priv/het		C	A	C/A	C/C	C/C	C/A	0	PDE4A
chr19	36234610	close	priv/het		G	A	G/A	G/G	G/G	G/G	-	U2AF1L4
chr19	55107829	control	priv/het		C	A	C/A	C/A	C/A	C/C	0	LILRA1
chr20	35234357	close/ family	priv/het		G	T	G/T	G/G	G/G	G/G	-	C20orf24
chr22	30742434	control	priv/het		T	C	T/C	T/C	T/T	T/C	0	SF3A1

Chr	Pos	From	Type	SNP	Ref	Alt	Boy	Grandfather mother's side	Grandmother father's side	Grandfather father's side
chr1	17085995	close/ family	priv/het		G	GC	G/GC	-	G/G	G/G
chr1	109650648	family	priv/het		A	T	A/T	-	-	A/A
chr1	207758220	close	priv/het		TG	T	TG/T			
chr2	69659126	close/ family	hom	rs4453725	A	T	T/T	A/A	A/T	A/A
chr2	135675908	close	hom	rs1348793	T	C	C/C	C/C	T/T	T/T
chr3	46414696	control	priv/het		T	A	T/A	T/T	T/T	T/T
chr5	149563573	close/ family	priv/hom	rs73796347	A	G	G/G	-	A/A	-
chr10	99376152	close	hom	rs3814556	A	G	G/G	A/G	A/A	G/G
chr14	21458320	close/ family	Hom	rs147038155	CCCCG	C	C/C	CCCCG/C	CCCCG/C	CCCCG/C
chr14	64855040	control	priv/het		CT	C	CT/C	CT/CT	CT/C	CT/CT
chr15	49867232	control	priv/het		C	T	C/T	C/C	C/T	C/C
chr17	41063408	control	priv/het		C	T	C/T	C/C	C/C	C/T
chr17	45567593	close/ family	hom		ATC	A	A/A	ATC/ATC	ATC/A	ATC/A
chr19	6897464	close	hom	rs330880	C	G	G/G	G/G	C/G	C/G
chr19	10577801	control	priv/het		C	A	C/A	C/C	C/A	C/C
chr19	36234610	close	priv/het		G	A	G/A	G/G	G/A	G/A
chr19	55107829	control	priv/het		C	A	C/A	C/C	C/C	C/C
chr20	35234357	close/ family	priv/het		G	T	G/T	G/G	G/G	G/G
chr22	30742434	control	priv/het		T	C	T/C	T/T	T/C	T/T

a)



b)

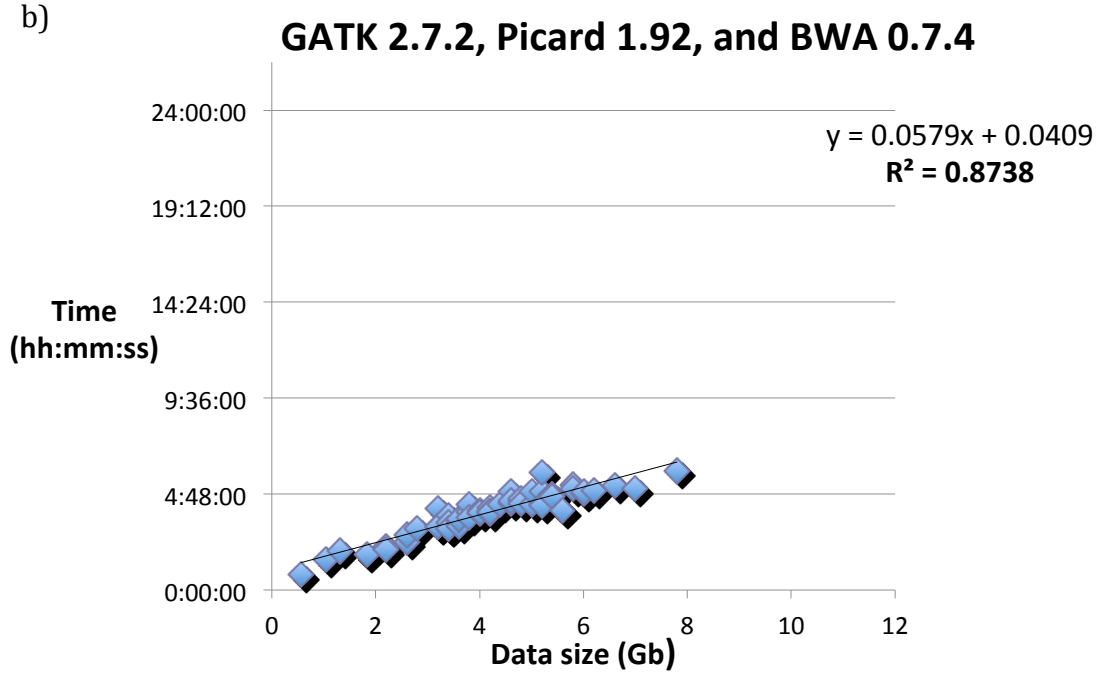


Figure S1 | Time estimate
The two graphs show the time estimation for running a sample with finalbam.sh, from fastq to final BAM. a) shows the time estimate for the first try with finalbam.sh while b) shows the time estimation after changing the versions of the programs.