Amino acid variation in human proteome

Vuković, Kristijan

Master's thesis / Diplomski rad

2016

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Science / Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:217:538899

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

Repository / Repozitorij:

Repository of the Faculty of Science - University of Zagreb

Kristijan Vuković

Amino acid variation in human proteome

Master's Thesis

Submitted to the Department of Chemistry,

Faculty of Science, University of Zagreb,

for the academic degree of

Master of Science in Chemistry

Zagreb

2016.

This master's thesis was written at the Centre for Systems Genomics, University of Melbourne under the mentorship of Assoc. Prof. Michael Inouye (University of Melbourne) and Assoc. Prof. Tomica Hrenar (University of Zagreb).

I would like to thank...

My mentors, prof. Inouye and prof. Hrenar for trust, support and advice during the course of this research. I couldn't have wished for a better team.

Mike's lab group at the Centre for Systems Genomics for accepting me in their midst and helping me settle in Melbourne. And especially Liam for his patience with my work on supercomputing cluster, as well as Artika and Lesley who were always there for a chat.

The whole Department of physical chemistry at the University of Zagreb for flexibility and help in organizing this research and especially to prof. Bertoša who thought me a lot. Also, the bioinformatics team at Molecular Biology department for making it as simple as possible to take their courses.

Karlo, Goja, Maja and Jana for all our crazy gaming nights and for making me chill out every now and then (and now and then...) Also, Nina, Petar and Iva for being reliable friends throughout our entire studies.

Lucy, for turning my life upside down and for taking part in all these adventures. Everything is easier beside you. And Grga, for everything.

My brother Viktor, mom and dad for being immensely supportive and understanding.

§ Table of Contents

§ 1. Introduction 1

§ 2. Literature review 3

2.1. Amino acids and proteins 3

2.1.1. Human proteome 3

2.1.2. Structure and Classification 6

2.2. Genetic code 8

2.2.1. Degeneracy 10

2.2.2. Mutations 10

2.2.3. Disease association 12

2.3. Sequencing methods 13

2.3.1. Next Generation sequencing 15

2.3.2. 3rd generation sequencing and nanopore 19

2.4. 1000 Genomes Project 21

§ 3. Theoretical background 25

3.1. Probability distributions 25

3.1.1. Continuous uniform distribution 26

3.1.2. Normal distribution 27

3.1.3. Poisson distribution 29

3.1.4. χ^2 distribution 30

3.2. Statistical hypothesis testing 31

3.2.1. Null hypothesis vs. alternative hypothesis 32

3.2.2. Test statistics and p-value 33

3.2.3. Decision Errors 34

3.3. Bioinformatics 36

3.4. Data 37

§ 4. Results and discussion 39

4.1. Map of amino acid substitutions 39

4.2. Disease causing variants 41

4.3. Pathogenicity and occurrence frequencies of individual amino acids 43

4.4. Maps based on structural classification 47

4.5. Codons in the genetic code influence the amino acid distributions 49

4.5.1. Codon position significance 51

4.5.2. Variants with multiple nucleotide substitutions 53

4.5.3. Synonymous variants 54

4.6. SIFT and PolyPhen 54

4.6.1. Statistical tests 55

4.6.2. Comparison between methods *57*
4.7. Population analysis **58**
4.7.1. Between population variations *59*
4.7.2. Within population variations *61*
4.8. Disease association of amino acids with combined substitution order **62**
4.9. Structural analysis of Trp ⇒ Ser substitutions **64**
4.9.1. Simulated proteins *65*
4.9.2. MD simulations *66*
4.9.3. Initial analysis *67*
§ 5. Conclusion **70**
§ 6. Materials and methods **72**
§ 7. Extended data **80**
§ 8. Literature **87**
§ 9. Supplements **90**
9.1. Code for programming language R **90**
9.2. List of symbols and abbreviations **101**
§ 10. Biography **102**

University of Zagreb

Faculty of Science

Department of Chemistry

Master's Thesis

Amino acid variation in human proteome

Kristijan Vuković

Centre for Systems Genomics, University of Melbourne
184 Royal Parade, Parkville Victoria 3052, Australia

Amino acids are important biological molecules that influence human health and disease through their function as structural units of proteins. Recent advancements to the genome sequencing technologies enabled an indirect, large-scale exploration of human proteomic diversity by mapping of nucleotide sequences in protein coding genomic regions to their corresponding amino acid sequences in proteome. In this thesis, data from the 1000 Genomes and several other sequencing projects was used for the construction of amino acid substitution maps that explore their occurrence frequency and pathogenicity. Biochemical structural classification of amino acids was identified as an important element in predicting both of these, with most class transitions showing the inversely proportional relationship between the two. Codon distribution of the underlying genetic code was compared with substitution maps and proved insufficient to account for the observed frequencies. Classification accuracy of pathogenic variants from SIFT and PolyPhen, two bioinformatical tools used in the 1000 Genomes Project was assessed, and frequency resolved substitution maps for 5 population groups defined in the project were constructed and analyzed. Finally, Trp ⇒ Ser, a substitution that continuously showed high pathogenicity signal was further explored through a series of molecular dynamics simulations.

(103 + XVIII pages, 37 figures, 7 tables, 85 references, original in English)

Thesis is deposited in Central Chemical Library at Horvatovac 102a, Zagreb

Keywords: 1000 Genomes / bioinformatics / human proteome / molecular dynamics / sequencing technologies / statistical hypothesis testing

Mentors: Assoc. Prof. Michael Inouye
Assoc. Prof. Tomica Hrenar

Reviewers: 1. Assoc. Prof. Tomica Hrenar
2. Prof. Zlatko Mihalić
3. Doc. Marko Močibob

Thesis accepted: 23.09.2016

SAŽETAK

Sveučilište u Zagrebu

Prirodoslovno-matematički fakultet

Kemijski odsjek

Diplomski rad

Aminokiselinska raznolikost ljudskog proteoma

Kristijan Vuković

Centre for Systems Genomics, University of Melbourne
184 Royal Parade, Parkville Victoria 3052, Australia

Aminokiseline su važne biološke molekule koje kao monomerne jedinice sudjeluju u izgradnji proteina. Nedavni razvoj tehnologija za sekvenciranje genoma omogućio je indirektno određivanje njihovih sekvenci mapiranjem nukleotidnih sljedova protein-kodirajućih regija genoma na odgovarajuće aminokiselinske sljedove proteoma. Takva metoda višestruko je brža i jeftinija od direktnog sekvenciranja proteina, a nove tehnologije revolucionarizirale su primjenu genetskog koda u medicinskim i znanstvenim istraživanjima. U ovom radu, podatci iz projekta "1000 Genomes" i nekoliko sličnih projekata iskorišteni su za konstrukciju mapa aminokiselinskih supstitucija s obzirom na njihovu učestalost i patogenost. Biokemijska klasifikacija aminokiselina pokazala se ključnom u objašnjavanju dobivenih rezultata, a tranzicije između strukturnih grupa u ove dvije mape imale su značajnu obrnuto proporcionalnu korelaciju koja je u skladu s evolucijskim selektivnim pritiscima. Distribucija kodona pojedinih aminokiselina pokazala se nedostatnom za objašnjavanje dobivenih supstitucijskih mapa, a s obzirom na nukleotidne pozicije unutar kodona, uočena je povećana patogenost aminokiselinskih supstitucija uzrokovanih promjenom 2. te smanjena patogenost supstitucija uzrokovanih promjenom 3. nukleotida. Testirana su dva bioinformatička alata za klasifikaciju patogenosti aminokiselinskih supstitucija korištena u projektu "1000 Genomes" te se alat PolyPhen pokazao nešto boljim od SIFT-a u detekciji patogenih supstitucija. Također, konstruirane su i analizirane frekvencijski razriješene supstitucijske mape 5 populacijskih grupa definiranih u tom projektu. Konačno, supstitucija Trp $\Rightarrow$ Ser, koja je u više analiza pokazala značajan signal patogenosti, detaljnije je strukturno analizirana kroz seriju molekulsko dinamičkih simulacija. Simulirani su proteini u kojima je dotična supstitucija detektirana i to uz pouzdanu klasifikaciju patogenosti.

(103 + XVIII stranica, 37 slika, 7 tablica, 85 literaturnih navoda, jezik izvornika: Engleski)

Diplomski rad pohranjen je u Središnjoj kemijskoj knjižnici na adresi Horvatovac 102a, Zagreb

Ključne riječi: 1000 Genomes / bioinformatika / humani proteom / molekulska dinamika / sekvenciranje genoma / statističko testiranje hipoteza

Mentori: Izv. prof. dr. sc. Michael Inouye
Izv. prof. dr. sc. Tomica Hrenar

Ocjenitelji:
1. Izv. prof. dr. sc. Tomica Hrenar
2. Prof. dr. sc. Zlatko Mihalić
3. Doc. dr. sc. Marko Močibob

Rad prihvaćen: 23.09.2016

Uvod

Nedavni razvoj tehnologija za sekvenciranje genoma, *tzv.* "metoda sekvenciranja sljedeće generacije" omogućio je prikupljanje velike količine podataka o nukleotidnim sljedovima u ljudskom genomu. Ti su podatci uspješno iskorišteni za brojna medicinska i znanstvena istraživanja što je dodatno potaknulo razvoj ovih tehnologija, ovaj put s istančanim zahtjevima o kvaliteti i dužini fragmenata koje je potrebno generirati. Danas se istražuje *tzv.* "treća generacija tehnologija sekvenciranja" koja kao cilj imaju drastično pojednostaviti cijeli postupak i dodatno skratiti vrijeme potrebno za sekvenciranje. Ovakav uzročno-posljedični proces eksponencijalno je povećao broj nukleotidnih baza spremljenih u online repozitorijima te obećava i dalje nastaviti dosadašnji trend. Međutim, ova stopa generiranja podataka postavila je njihovu bioinformatičku analiza kao limitirajući faktor mnogih istraživanja.

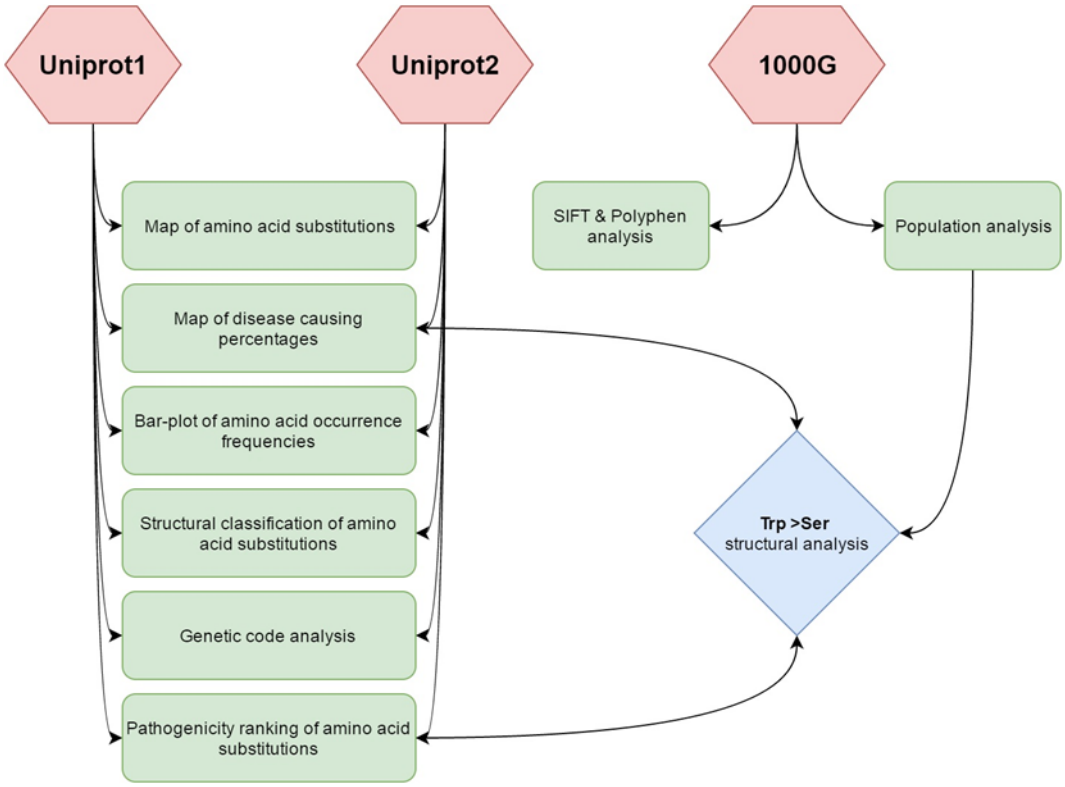
Jedan od nedavno završenih globalnih projekata sekvenciranja je projekt 1000 genoma čiji su konačni rezultati objavljeni prošli listopad, a koji je uključivao određivanje kompletne genomske sekvence i svih strukturnih varijacija 2504 osobe iz 5 populacijskih grupa. Kao i kod mnogih drugih projekata, rezultati su pohranjeni u javno dostupnom repozitoriju. Velika većina otkrivenih varijacija posljedica je SNP-ova, promjena individualnih nukleotida u genetskoj sekvenci. Ako SNP pogađa protein-kodirajuću regiju genoma, on neizravno utječe i na promjenu aminokiseline u primarnoj strukturi proteina translatiranog iz ove sekvence.

Cilj ovog rada je istražiti aminokiselinske supstitucije uzrokovane SNP-ovima u protein-kodirajućim regijama genoma (slika 0.1). U tu svrhu, bioinformatičke metode i alati iskorišteni su za konstrukciju supstitucijskih mapa s obzirom na učestalost i patogenost. Dobivene mape analizirane su prema tablici genetskog koda u kojoj je prikazana distribucija kodona, 3-nukleotidnih sljedova koji kodiraju pojedine aminokiseline. Mapa su također analizirane prema biokemijskoj klasifikaciji aminokiselina s obzirom na njihove strukturne elemente. Varijacije otkrivene u projektu 1000 genoma, kao i u drugim sličnim projektima, analizirane su i na razini pojedinih aminokiselina u referentnim i alternativnim proteinskim sekvencama te su konstruirani odgovarajući grafikoni njihovih učestalosti i patogenosti.

Nadalje, neki od rezultata samog projekta 1000 genoma zasebno su analizirani. Tako je određen postotak sinonimnih SNP-ovi, nukleotidnih supstitucija koje ne uzrokuju

promjenu u primarnoj proteinskoj strukturi, analizirani su rezultati 2 bioinformatička alata za predviđanje patogenosti aminokiselinskih supstitucija te su konstruirane frekvencijski razriješene supstitucijske mape za svaku od pet populacijskih grupa definiranih u projektu.

Konačno, započeto je detaljnije istraživanje aminokiselinske supstitucije Trp ⇒ Ser, koja je u više analiza pokazala značajan signal patogenosti, i to kroz seriju simulacija molekulske dinamike provedenih u programu GROMACS. Simulirani su svi proteini kod kojih je ova supstitucija imala pouzdanu klasifikaciju patogenosti i za koje je pronađena riješena kristalna struktura zadovoljavajuće kvalitete. Kod svih proteina simuliran je divlji tip te njeogova mutirana varijanta kod koje je triptofan zamijenjen serinom. Sve simulacije provedene su u trajanju od najmanje 30 ns, a pretraživane su konformacijske promjene između divljeg i mutiranog proteina do kojih dolazi u patogenim, a ne dolazi u benignim varijantama ove supstitucije.



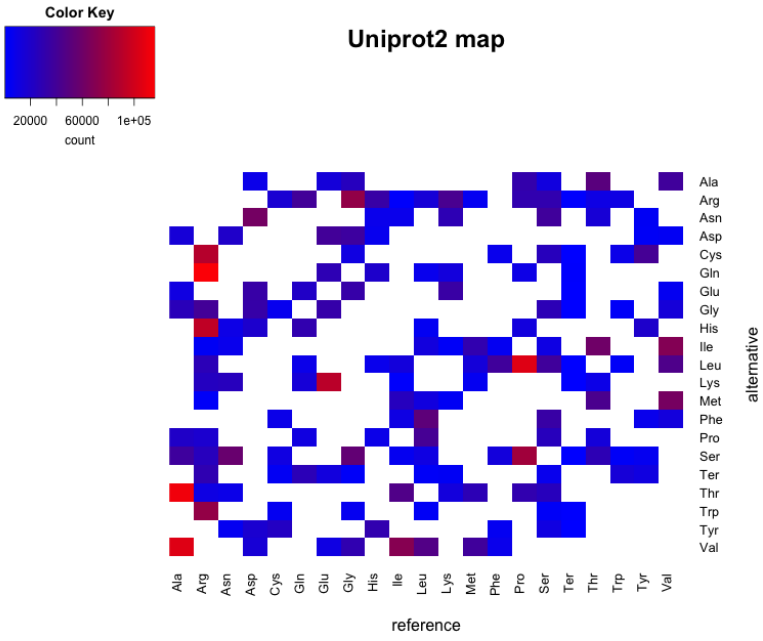
Slika 0.1: Shema analiza provedenih u sklopu ovog rada.

Rezultati i diskusija

Za odgovarajuće aminokiselinske varijante pronađene su i analizirane 3 baze podataka nazvane Uniprot1, Uniprot2 i 1000G. Uniprot1 je ručno pregledana baza poznatih

aminokiselinskih supstitucija iz raznih izvora. Bitno je napomenuti da ova baza sadrži i supstitucije koje nisu mogle nastati kao posljedica SNP-ova, jer zahtijevaju promjenu više od jednog nukleotida u kodonu referentne aminokiseline. Također, ova baza obogaćena je patogenim varijantama jer us one zanimljivije za daljnja istraživanja zbog čega su češće ručno pregledavane. Uniprot2 baza aminokiselinskih supstitucija nastala je mapiranje i *in silico* translacijom SNP-ova iz protein kodirajućih regija ljudskog genoma. Ova baza sadrži znatno veći broj supstitucija, ali njihove su klasifikacije patogenosti manje pouzdane jer unosi u bazi nisu ručno pregledani. 1000G set podataka su zapravo neprocesirani rezultati projekta 1000 genoma. Ovaj set sadrži neke dodatne podatke koji su izbačeni iz drugih baza, ali zato nema klasifikaciju patogenosti koja je za potrebe analize mapirana iz Uniprot1 baze.

Mapa supstitucijskih učestalosti konstruirana je za Uniprot1 i Uniprot2 baze. Iz njihovih sličnosti vidljivo je da Uniprot1, unatoč svojoj veličini, dobro opisuje ukupnu distribuciju varijanti. Prikazana je mapa dobivena iz Uniprot2 baze (slika 0.2). Treba imati na umu da apsolutne vrijednosti supstitucijskih učestalosti u ovim mapama nisu relevantne jer baze ne sadrže podatke o frekvencijama pojedinih varijanti, zbog čega možemo uspoređivati samo relativne vrijednosti između njih.

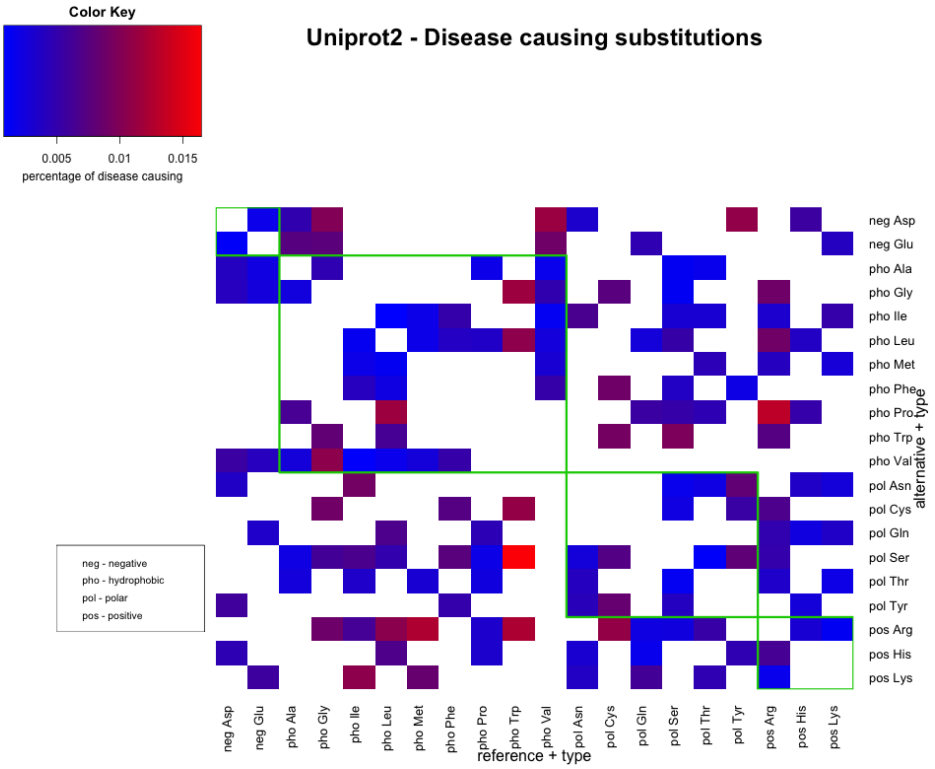


Slika 0.2: Mapa učestalosti aminokiselinskih supstitucija.

χ^2 test korišten je za usporedbu dobivenih mapa s mapom konstruiranom iz tablice genetskog koda. Nul hipoteza da dobivene učestalosti proizlaze iz distribucije koju predviđa

raspodjela kodona odbačena je s velikom pouzdanošću (p-vrijednost < 10^{-323}). Ovakav ishod je očekivan jer su biokemijski procesi transkripcije i translacije, kao i njihove evolucijske promjene, znatno kompleksniji od onoga što predviđa osnovni genetski kod.

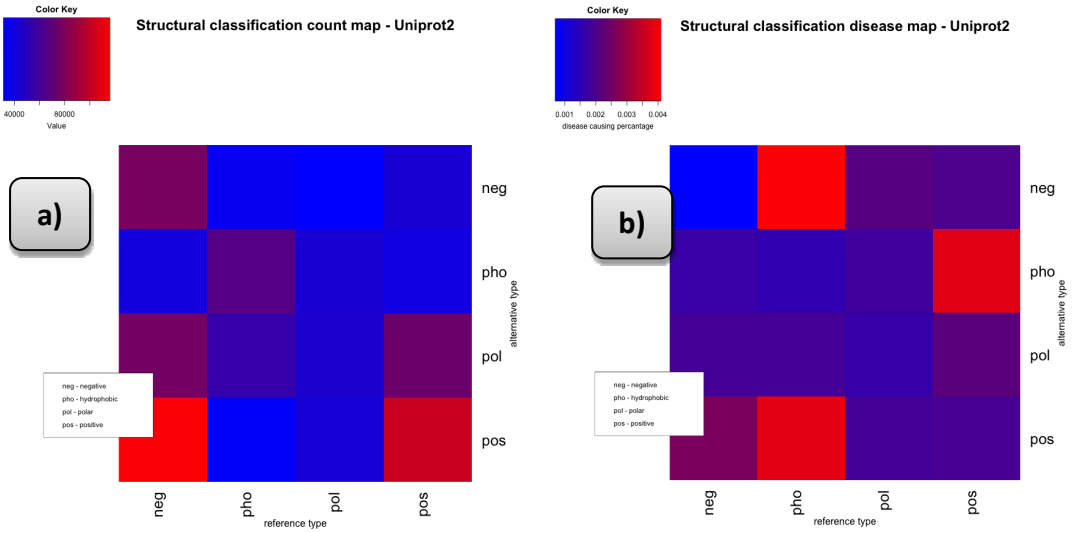
S obzirom na patogenost pojedinih supstitucija, mape postotka patogenih supstitucija konstruirane su za oba seta podataka. Prikazana je mapa dobivena iz Uniprot2 baze (slika 0.3) iz koje su supstitucije Trp ⇒ Ser, Arg ⇒ Pro i Cys ⇒ Phe detektirane kao najopasnije. Ovi su signali, kao i većina drugih, konzistentni s rezultatima iz Uniprot1. Bitno je uočiti da niti ovdje apsolutne vrijednosti signala nisu pouzdane. Za Uniprot1 one su prevelike jer je cijela baza obogaćena patogenim varijantama, dok su za Uniprot2 premale jer velikom postotku varijanti nedostaje klasifikacija te je određeni postotak ovih nedefiniranih varijanti također patogen. Relativne vrijednosti između supstitucija mogu se uspoređivati i konsistentne su između ove dvije baze što se posebno dobro vidi nakon normalizacije rezultata.



Slika 0.3: Mapa patogenosti supstitucija konstruirana iz Uniprot2 baze.

Zeleno ograđeno područje prikazane mape predstavlja supstitucije koje ne mijenjaju strukturni tip aminokiseline. Evidentno je da ove supstitucije imaju manji postotak

patogenosti što je očekivano s biokemijskog stajališta. Ovaj efekt detaljnije je analiziran na mapama strukturnih supstitucija u kojima su aminokiseline koje pripadaju istom tipu grupirane, a njihove učestalosti normalizirane prema broju aminokiselina unutar grupe kao i broju načina na koje do pojedinih supstitucija može doći. Ponovno su konstruirane mape učestalosti i patogenosti uz ovakvu klasifikaciju (slika 0.4). Dvije interesantne stvari vidljive su s mapa. Na mapi patogenosti, dijagonala ima najmanje vrijednosti što potvrđuje da se ove supstitucije najčešće toleriraju. Također, vidi se obrnuto proporcionalna korelaciju između dviju mapa. Ovo je najočitije kod pho $\Rightarrow$ pos, pos $\Rightarrow$ pho i pho $\Rightarrow$ neg supstitucija koje imaju najveći postotak patogenosti te su istovremeno najrjeđe po učestalosti, što se može objasniti evolucijskom negativnom selekcijom ovih tipova supstitucija.

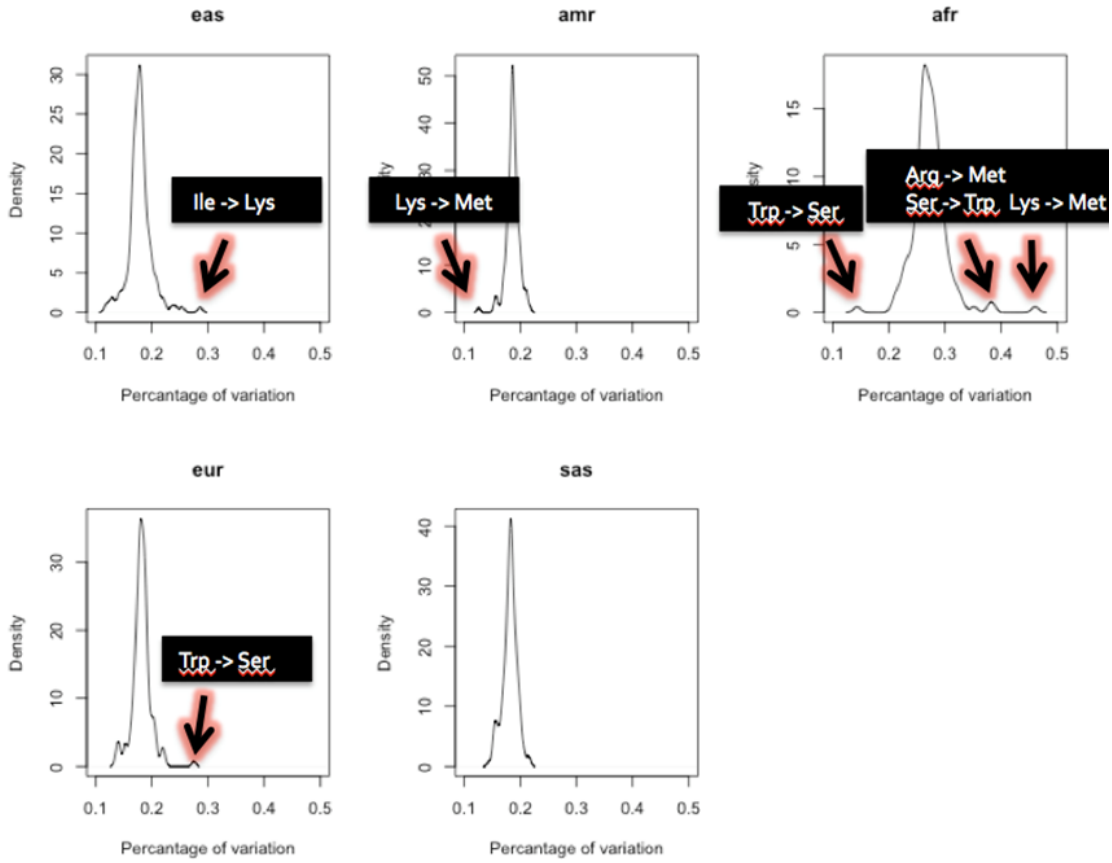


Slika 0.4: Supstitucijske mape bazirane na strukturnoj klasifikaciji aminokiselina. **a)** učestalosti, **b)** patogenosti.

Analiza je nastavljena promatranjem odnosa genetskog koda i dobivenih mapa pri čemu je uočena povećana patogenost supstitucija uzrokovanih SNP-om na 2. nukleotidu kodona te smanjena patogenost supstitucija kod kojih je SNP na 3. nukleotidu.

Set podataka 1000G iskorišten je za analizu dodatnih rezultata projekta 1000 genoma. Uočena je neravnomjernost broja sinonimnih supstitucija koja je također konsistenta s evolucijskom perspektivom, budući da su njihove učestalost znatno veće od očekivanih. Nadalje, analizirani su rezultati 2 bioinformatička alata za predviđanje patogenosti aminokiselinskih supstitucija te se alat PolyPhen pokazao nešto boljim od SIFT-a

u detekciji patogenih supstitucija. Također, konstruirane su i analizirane frekvencijski razriješene supstitucijske mape 5 populacijskih grupa definiranih u tom projektu. Kod ove analize uočena je povećana učestalost skoro svih varijacija u afričkoj populacijskoj grupi što je konzistentno s rezultatima projekta koji su uočili isti efekt u analizi broja strukturnih varijacija na razini cijelog genoma. Također je uočen zanimljiv odnos između povećane i smanjene učestalosti nekoliko supstitucija kod određenih populacijskih grupa (slika 0.5). Tako Lys ⇒ Met supstitucija ima povećanu učestalost u afričkoj, a smanjenu u američkoj populacijskoj grupi dok je Trp ⇒ Ser znatno učestalija u europskim genomima u odnosu na afričke. Ove razlike mogle bi biti posljedica nekih biokemijskih promjena između populacijskih grupa do kojih je došlo tijekom evolucije.



Slika 0.5: Distribucije postotka supstitucija u pojedinim populacijskim grupama.

Konačno, supstitucija Trp ⇒ Ser, koja je u više analiza pokazala značajan signal patogenosti, detaljnije je strukturno analizirana kroz seriju molekulsko dinamičkih simulacija. Simulirano je ukupno 20 proteina u kojima je dotična supstitucija detektirana i to uz pouzdanu klasifikaciju patogenosti. Za svaki protein simuliran je divlji tip i njegova

mutirana varijanta u kojoj je triptofan zamijenjen serinom te je započeta analiza dobivenih trajektorija pretragom konformacijskih promjena do kojih dolazi kod patogenih parova simulacija, a ne dolazi kod benignih.

Zaključak

U ovom radu detaljno su istražene aminokiselinske supstitucije koje se pojavljuju u ljudskom proteomu, a posljedica su SNP-ova u protein-kodirajućim regijama genoma. Analizirani su rezultati projekta 1000 genoma, koji su bili glavna motivacija za istraživanje, te rezultati nekolicine sličnih projekata. Aminokiselinska supstitucija Trp $\Rightarrow$ Ser odabrana je za daljnju strukturnu analizu kod koje će trajektorije dobivene molekulsko dinamičkim simulacijama biti uspoređivane između patogenih i benignih varijanti ovih supstitucija.

§ 1. Introduction

Genome sequencing technologies provide immense potential for understanding scientific and medical underpinnings of human genetic code. The rate at which current technologies generate sequencing data and the rate at which they are being improved promise to continue this trend. However, this progress also makes it hard for bioinformatical analysis to keep pace. Several large-scale sequencing projects have provided extensive data that has been either just partially analyzed or simply filtered and uploaded to one of the numerous online repositories. An example of most recently completed one is the 1000 Genomes Project. The aim of this thesis is to explore some of the results of this and several other sequencing projects. Primary objective is analyzing amino acid substitutions that arise as consequences of single-nucleotide polymorphisms (SNPs, pronounced: *snips*) in the protein-coding regions of the human genome.

To accomplish this, bioinformatical tools and methods are applied on amino acid structural classification from biochemical models and on other relevant chemical and biological parameters such as molecular mass and underlying genetic code of amino acids. Statistical tests are used to assess the significance of the results. Similar analysis with a smaller dataset was published in 2003.,[1] but only these recent sequencing projects enable completely unbiased and genome-wide exploration of amino acid variants.

First part of this thesis provides a review of relevant biochemical concepts used during the analysis as well as a short summary of the 1000 genomes project whose results were the main motivation for this research (although not the only ones used). Then, the mathematical background for the statistical hypothesis testing and few other theoretical topics is provided. This is followed by the main section in which research results are discussed. Some more complicated calculations and procedure descriptions from this part are given separately in the Materials and methods section. Also, some overly detailed and less revealing figures are provided separately in the Extended data section. Finally, short conclusion of the research objectives and computational code for the analysis are given.

The main section, in which research results are discussed, consists of the analysis conducted on three datasets: Uniprot1, Uniprot2 and 1000G (figure 1.1.) First research objective was constructing a map of all known amino acid substitutions. Then, the amino acid substitutions were explored with respect to their disease association: Uniprot1 and

Uniprot2 datasets provide these classifications, but they are incomplete in both. In Uniprot1 the classification is biased towards pathogenic variants and in Uniprot2 it's incomplete because for the majority of variants the classification is unknown. Regardless of this shortcoming, the unbiased Uniprot2 dataset is preferred for most analysis, as will be discussed further (Materials and methods). Then, the two previous results were analyzed with respect to the individual amino acids, their structural classification and their underlying genetic code. These research topics were supplemented with analysis of the unprocessed 1000 Genomes Project results in the 1000G dataset. Pathogenicity predictions of two bioinformatical tools were assessed and amino acid substitution maps constructed for five population groups whose genomes were sequenced in this project. Finally, by taking several interesting results into account, disease ranking table of amino acid substitutions was constructed.

Last research objective of this thesis was structural analysis of Trp ⇒ Ser mutation. Structural analysis was carried out through a series of molecular dynamics (MD) simulations in program package GROMACS.[2] The setup of MD simulations and the analysis procedures are described. This part of the research is still in progress and only preliminary results are shown in the thesis.

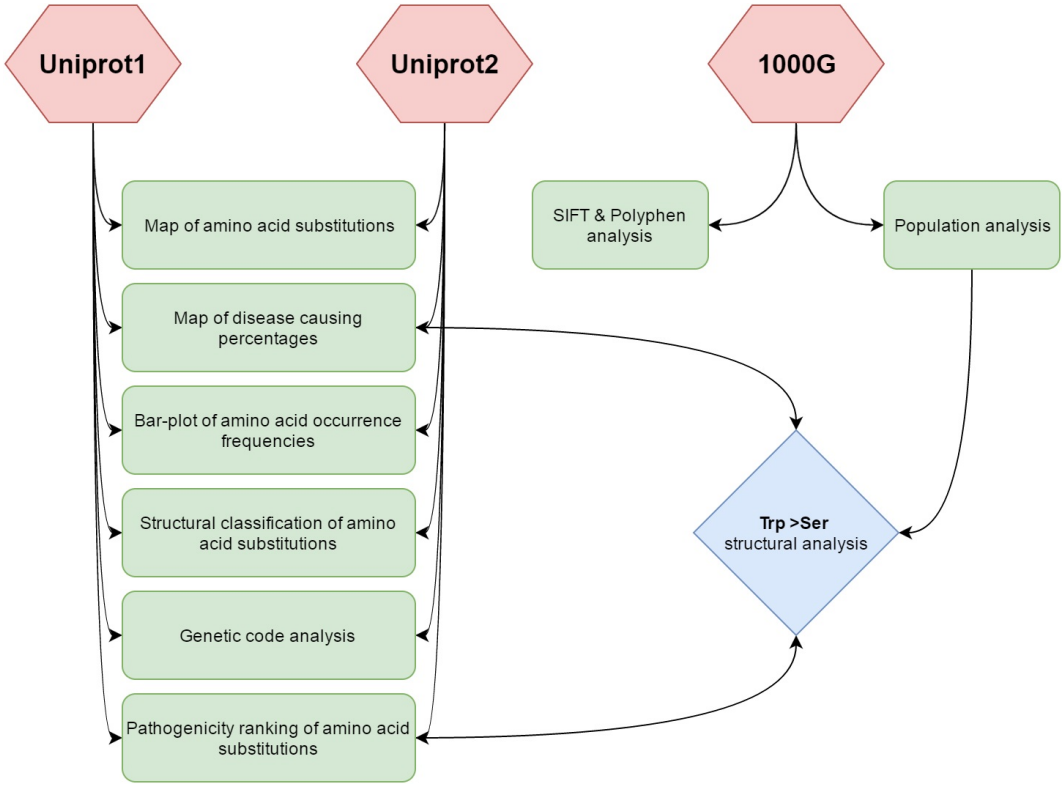


Figure 1.1: Flowchart of the analysis presented in this thesis.

§ 2. Literature review

2.1. Amino acids and proteins

Amino acids are organic compounds that contain amine (NH_2) and carboxylic acid (COOH) functional groups. All amino acids share fundamental structural elements and differ in chemistry of their respective side-chains. There are about 500 naturally occurring amino acids[3] and they can be classified in various ways. Chemical structure guides classification in accordance to the location of core structural group and defines them as alpha (α-), beta (β-), gamma (γ-) or delta (δ-). In terms of physicochemical properties, they can be classified based on their side-chain structure. This also, to large extent, influences their biochemical characteristics. Additionally, they can be classified in relation to their occurrence frequency, metabolic production and finally, protein building capacity, which is their most important biological property.

Twenty-two amino acids occur naturally as structural units of proteins and are therefore called proteinogenic or natural amino acids. Twenty of these are also encoded by the universal genetic code (see chapter 2.2.) Selenocysteine and pyrrolysine are in the first group but not in the second and they have separate synthetic mechanisms for protein incorporation.[4][5] For this reason, they occur less frequently in the human proteome. To function as monomer units, amino acids form polymer chains through the peptide bond formation process (figure 2.1). In this condensation reaction, C-terminus of first, and N-terminus of second amino form the peptide bond and the water molecule is released. This type of chemical reaction occurs continuously in all living systems thus synthesizing polypeptides or proteins.

2.1.1. Human proteome

Proteome is a term that refers to the complete set of proteins expressed by a genome, cell, tissue, or organism at a certain time. The term was coined by Marc Wilkins[6] as an analogy to the term genome which refers to the complete genetic sequence of an organism. The human proteome consists of approximately 70,600 proteins.[7] There is an ongoing effort to determine this precise number and map all of them.[8] As there are about 20,500 genes in the human genome, depending on the exact definition of the gene[9], it follows that a

significant part of our proteome arises from alternative splicing events. The number of protein coding genes was estimated a lot higher. It was argued that explaining the human complexity, compared to that of other organisms, requires larger protein diversity.[10] Now we know there are other functional elements that account for this.[11] Over the whole protein coding sequence of the human genome, mean exon size is 145 bp (median 122 bp) with a mean of 8.8 per gene (median 7), corresponding to a mean protein size of 447 amino acids (median 367). [12]

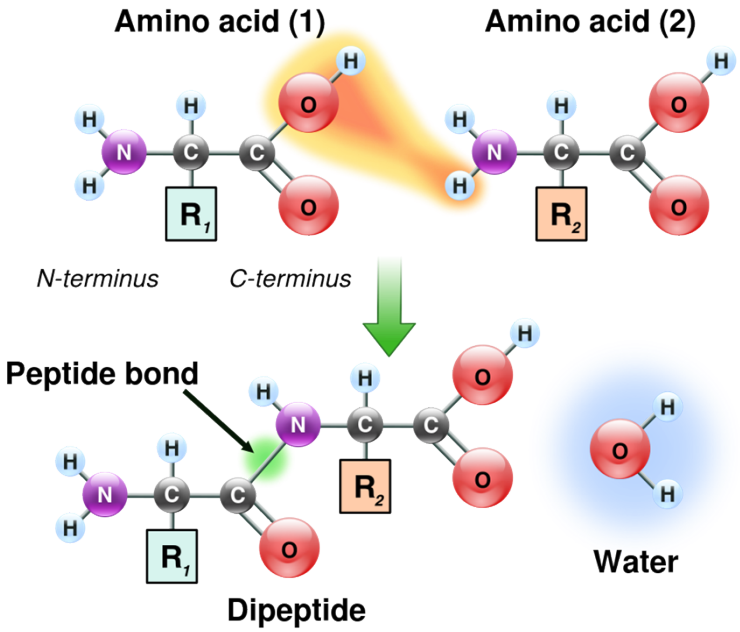


Figure 2.1: α- amino acid structure and peptide bond formation.[13]

There are 4 layers of information that define each protein. Primary structure refers to the sequence of amino acids in the polypeptide chain. In this notation, residues are always counted from N- to C- terminus. Any post-translational modifications that amino acids undergo (*e.g.* acetylation, formylation, glycosylation *etc.*) are also considered a part of the primary structure, as are any disulfide bonds between cysteine residues. In general, primary structure corresponds to the information about protein elements that are held together by covalent bonds. Secondary structure is the local, three-dimensional sub-structure of polypeptide segments. For the most part, it is defined by patterns of hydrogen bonds between amino hydrogen and carbonyl oxygen atoms in the protein backbone. The most frequent elements of the secondary structure are alpha-helices, beta-sheets and

various turns and the easiest ways to determine and visualize them is on the Ramachandran plot. Ramachandran plot is a diagram of backbone dihedral angles between C-α and C (ψ) and N and C-α (ϕ) atoms. Plotted structural elements can be compared with the theoretically defined outline of their corresponding regions. This can be used to predict the secondary structure of a protein segment or to assess the quality of the experiment used to obtain it (*e.g.* homology modeling or X-ray crystallography, figure 2.2). Tertiary structure is the overall shape of a single protein molecule. It arises from the spatial relationship of secondary structures and is largely determined by non-local, Van der Waals interactions. Among these, hydrophobic effects were shown to have a predominant influence.[14] Finally, for oligomeric proteins, quaternary structure becomes relevant. This is a three-dimensional arrangement of multiple subunits that form the protein complex. Thus, it determines the spatial relationship of individual polypeptide molecules. Monomeric proteins don't have the quaternary structure.

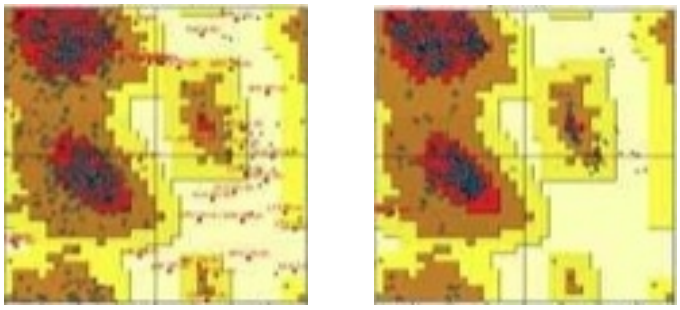


Figure 2.2: Ramachandran plot for **a)** low resolution (2.9 Å) crystal structure and **b)** high resolution (1.8 Å) crystal structure.[15]

Various experimental methods can be used to determine different layers of the protein structure. For the primary structure, Edman degradation is applicable.[16] This is, effectively, a peptide sequencing method in which chemical degradation is used to cleave a single residue from the N-terminus of the protein in each step of the procedure. Alternatively, protease enzymes can be used to cut the amino acid sequence at specific cleavage sites and mass spectroscopy, followed by database fingerprinting, to identify the obtained polypeptide segments. However, with the intense development of genome sequencing technologies (see chapter 2.3.), the simplest way to determine primary protein structure became mapping and *in silico* translation of the genomic region corresponding to

the protein of interest. This approach can be further extended to determine the variants in the primary structure by comparison with the reference amino acid sequence. The secondary structure of a protein can be easily assessed with circular dichroism. Although this method doesn't assign structural elements to the amino acid sequence unambiguously, it is a simple procedure that gives their overall content (*e.g.* 30 % alpha-helicies, 40 % beta-sheets, 30 % the rest). Precise determination of the secondary, as well as tertiary and quaternary structure, requires X-ray crystallography or NMR spectroscopy. These methods provide full three-dimensional protein conformation. NMR is simpler to perform but produces lower resolution of the protein structure. X-ray is the most common method and can produce resolution on the atomic level. However, it requires crystallized protein molecules, which can be very hard to obtain for many (especially transmembrane) proteins. Since this is often the best tool for structural research, a lot of effort has been invested into protein crystallization, even attempting crystal growth in the micro-gravitational environment.[17][18] Another method for determining protein structure, which emerged recently and is becoming increasingly popular is cryo-electron microscopy (cryo-EM).[19] Its application for structural biology was enabled by improvement of the resolution of cryo-EM maps. Main advantages of this method are that it doesn't require crystallized protein, since it can be applied on the highly purified protein solution, and that the structure it provides is the closest we can currently get to the conformation which protein has in its active state. This conformation always gets slightly distorted due to sample preparation steps and cryo-EM has least of those.

2.1.2. Structure and Classification

Amino acid classification that will be considered hereafter is the one based on their physicochemical properties and it will now be explored in greater detail. There are several ways in which residues* can be grouped together, but all methods focus on characteristics which can influence secondary and tertiary protein structure, *e.g.*, charge, size, hydrophobicity *etc.* Overall, classifications are pretty similar. Two structural groups that are consistent throughout all of them are acidic and basic amino acids (or negatively and

* In biochemistry the term residue refers to a specific monomer within the peptide or nucleic acid polymer and it will be used in this way.

positively charged, based on their charge in neutral pH environment). Aspartate and glutamate are acidic residues and lysine, arginine and histidine belong to the basic group. Furthermore, all classifications introduce polar and nonpolar (or hydrophobic) amino acids. Serine, threonine, cysteine, asparagine and glutamine are always considered polar while glycine, alanine, valine, leucine, isoleucine and methionine belong to hydrophobic group. After this, classification becomes more ambiguous. Some methods introduce additional group for residues with aromatic side-chains.[20] These residues are tryptophan, tyrosine and phenylalanine. Alternatively, tryptophan and phenylalanine may be added to the hydrophobic, and tyrosine to the polar amino acid group.[21] Finally, amino acids that don't strictly belong to their corresponding class (due to their specific characteristics or function) might be merged together into separate group of unclassified residues.[22] These are histidine, due to its small acidity relative to other acidic amino acids, cysteine, due to disulfide bond formation and proline, due to its unconventional structure. Classification chosen for subsequent analysis is the one based on 4 primary classes - positive, negative, hydrophobic and polar (figure 2.3). This classification introduces the smallest number of groups, which increases robustness of applied statistical methods (in a sense that statistical power of any potential signal won't be reduced by a small number of corresponding amino acids, see chapter 3.2.)

One more relevant partition of twenty proteinogenic amino acids is the one based on their metabolic production capacity. In this sense, 11 amino acids are considered non-essential since they can be synthesized by the organism, while the remaining 9 are called essential and they have to be obtained through nutrition. Histidine is, for the most part, a non-essential amino acid, but increased requirements during accelerated growth periods make it essential. The other 8 essential amino acids are: isoleucine, leucine, lysine, methionine, phenylalanine, threonine, tryptophan, and valine.[23]

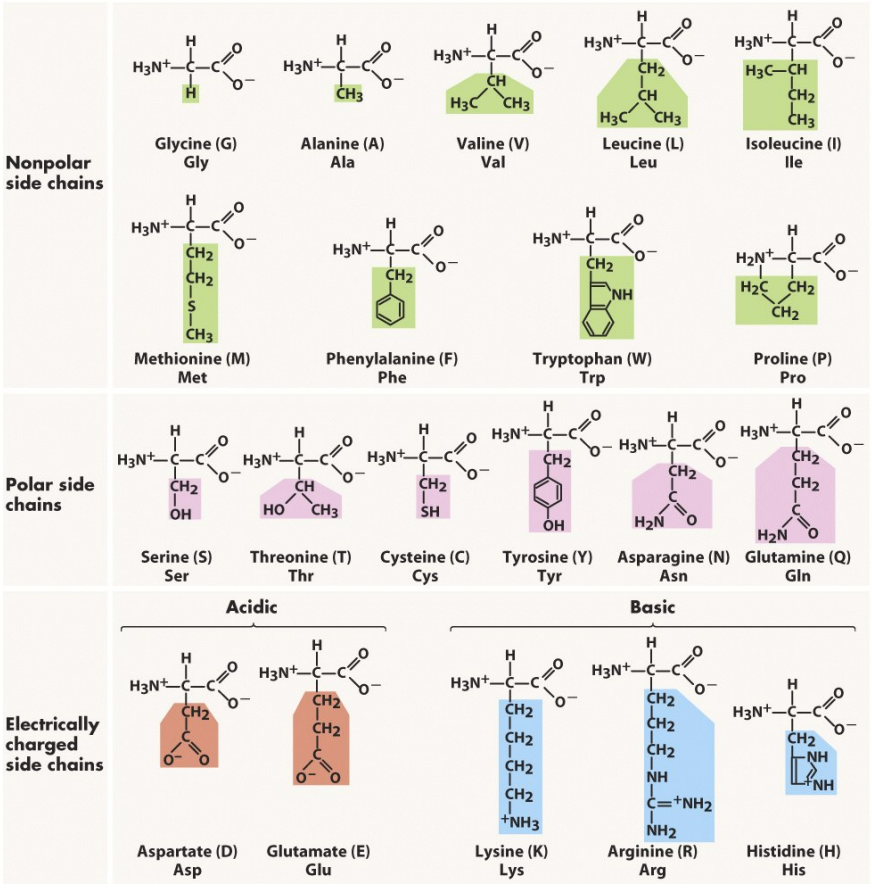


Figure 2.3: Amino acid classification used for the analysis.[21]

2.2. Genetic code

The genetic code is the set of rules by which information encoded within genetic material (DNA or mRNA sequence) is translated into proteins. The code defines how nucleotide triplets, called codons, specify amino acids that are incorporated during protein synthesis. Nucleotides are organic molecules that serve as monomers of nucleic acid chains. They consist of D-2-deoxyribose or D-ribose monosaccharide in DNA or RNA respectively, one of five nitrogenous bases and a phosphate group. Three out of five bases - adenine (A), guanine (G) and cytosine (C) occur in both DNA and RNA. Thymine (T) occurs in DNA and is unambiguously replaced by uracil (U) in RNA. Nitrogenous bases are the defining element of each nucleotide, which is why we often abbreviated the nucleotide monomers in DNA or RNA chain with starting letter of their corresponding base.

Four different nucleotides constitute both DNA (ACGT) and RNA (ACGU) and therefore allow for 64 different codons in each of them. In DNA molecule, nucleotides are

paired: A with T and C with G, forming two strands. During transcription, RNA polymerase adds complementary RNA nucleotides to a template DNA strand (with U unambiguously replacing T). The formed RNA strand is therefore identical to the second, coding DNA strand (again, with U replacing T, figure 2.4). It follows that codons defined from coding DNA strand are equivalent to the ones defined from RNA molecules, with every T replaced by U. The genetic code table, which lists amino acids corresponding to each codon, can therefore be represented in two equivalent ways, with RNA or DNA nucleotides. An RNA codon table is shown below and will be used for future reference (figure 2.5).

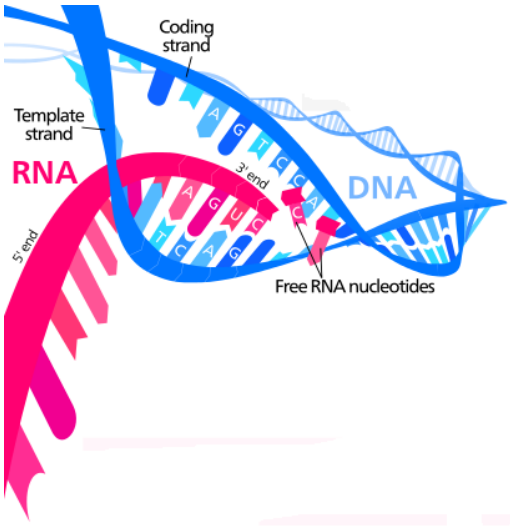


Figure 2.4: RNA transcription.[24]

Second letter

First letter	U	C	A	G	Third letter
U	UUU Phe UUC Phe UUA Leu UUG Leu	UCU Ser UCC Ser UCA Ser UCG Ser	UAU Tyr UAC Tyr UAA Stop UAG Stop	UGU Cys UGC Cys UGA Stop UGG Trp	U C A G
C	CUU Leu CUC Leu CUA Leu CUG Leu	CCU Pro CCC Pro CCA Pro CCG Pro	CAU His CAC His CAA Gln CAG Gln	CGU Arg CGC Arg CGA Arg CGG Arg	U C A G
A	AUU Ile AUC Ile AUA Ile AUG Met	ACU Thr ACC Thr ACA Thr ACG Thr	AAU Asn AAC Asn AAA Lys AAG Lys	AGU Ser AGC Ser AGA Arg AGG Arg	U C A G
G	GUU Val GUC Val GUA Val GUG Val	GCU Ala GCC Ala GCA Ala GCG Ala	GAU Asp GAC Asp GAA Glu GAG Glu	GGU Gly GGC Gly GGA Gly GGG Gly	U C A G

Figure 2.5: Genetic code table.

2.2.1. Degeneracy

Degeneracy of codons is the redundancy of the genetic code. There are 20 standard amino acids that build human proteome, but at the same time there are 64 codons that define them. Consequently, some amino acids must be defined by multiple codons (figure 2.5). A position of a codon is said to be an n-fold degenerate site if n (out of four) possible nucleotides at this position specify the same amino acid. If any mutation at this position results in amino acid substitution, a position of a codon is said to be a non-degenerate site. Degeneracy contributes to the robustness of the genetic code - if an error occurs in the protein-coding region of the genome or is introduced into mRNA during transcription, it can end up having no effect on the primary protein structure after translation.

During translation, tRNA molecules are required to pair mRNA codons with their corresponding amino acids. This is achieved through the anticodon-codon interaction. Each tRNA molecule contains anticodon - three nucleotides long sequence that is complementary to its corresponding mRNA codon and can therefore recognize it. Since there are 64 different codons, this would require 64 different tRNA molecules to be maintained in cells at all times. This is not energetically favorable and therefore, most organisms have fewer than 45 species of tRNA.[25] Evidently, each tRNA must be able to recognize more than one codon, which means that the original Watson-Crick pairing must be modified or additional rules introduced to account for these interactions. Alternative pairing, "Wobble hypothesis", was purposed already in 1966 by Francis Crick.[26] It was based on the observation that third codon base consistently has the highest degeneracy and suggested that some tRNA 5' anticodon bases recognize multiple mRNA 3' codon bases. Although Crick's wobble base pair rules subsequently got revised, their initial premise was upheld.[27] It is interesting to note how degeneracy of the genetic code, which at first seemed like a curios coincidence, turned out to be the consequence of evolutionary fine-tuning required to balance the robustness of the code with the optimization of cellular content.

2.2.2. Mutations

Mutations are permanent alterations of the nucleotide sequence of the genome. They can be roughly classified in two groups: small-scale and large scale. Small-scale mutations affect genome in only one or a few nucleotides while large-scale mutations spread across larger

genomic regions. Point mutations are subgroup of small-scale mutations that cause substitution of a single nucleotide. These are also called single-nucleotide polymorphisms and can be further classified as:

- Silent mutations: they occur outside of the protein-coding regions or within them but with no effect on amino acid sequence (due to degeneracy of the genetic code)
- Missense mutations: they occur in protein-coding region and cause amino acid substitution
- Nonsense mutations: they occur in protein-coding region and cause premature stop codon

Terminology for SNPs is somewhat different and they are classified as:

- Noncoding SNPs: they occur outside protein-coding regions
- Synonymous SNPs: they occur in protein-coding region but don't cause amino acid change
- Nonsynonymous SNPs (nsSNPs): they occur in protein-coding region and change the affected amino acid (into a different one or into a stop codon)

Mutations can have various effects on the fitness of the organism. Harmful mutations, also called deleterious, decrease the fitness of the organism and are under negative selection. Beneficial, or advantageous mutations increase the fitness and are consequently under positive selection. Neutral mutations have no harmful nor beneficial effect on the organism but are important in the neutral theory of molecular evolution as they provide basis for the genetic drift.[28]

Mutation rate is a measure of the rate at which various types of mutations occur over time. It's typically given for a specific class of mutations. Different genetic variants within a species are called alleles and so mutations are said to create new alleles. Point mutation rate for human genomic DNA has been estimated in the range from 1.1×10^{-8}[29] to 2.5×10^{-8}[30] per site per generation. However, there is a significant difference between rates for transitions and transversions (figure 2.6). Overall, two average human genomes currently differ at ~0.15 % positions of the 3 billion bp long genome sequence[31], but the nucleotide discrepancy is somewhat greater, since not all of these sites correspond to point-mutations (see chapter 2.4.)

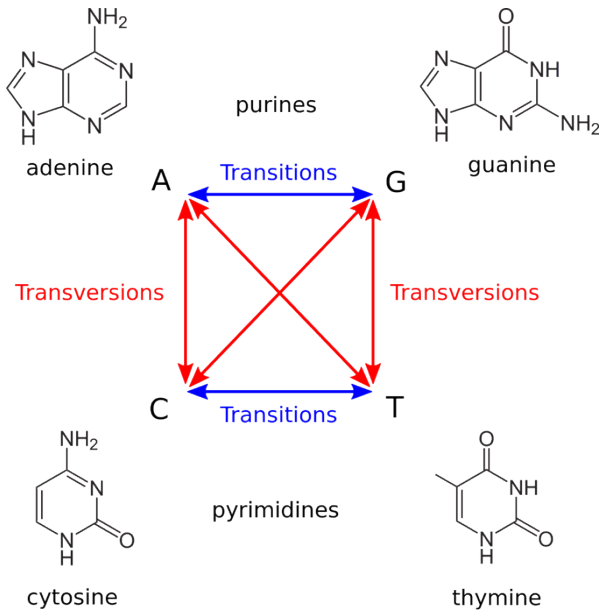


Figure 2.6: Transitions and transversions of nucleotides.[32]

2.2.3. Disease association

Phenotype is the composite of an organism's characteristics and traits. It's the result of information stored in the organism's genome combined with the environmental factors. Some traits are more influenced by one aspect, some by the other. This is often called nature (genome) vs nurture (environment) contribution. Mutations in the genome can therefore have a significant influence on the phenotype. If this effect is negative and contributes to a disease, the mutation is said to be disease causing or disease associated. There are many types of disease causing mutations and they affect the phenotype in different ways and with various degrees of disease association.

Mendelian diseases are, from a genetic perspective, the simplest type of a disease. They are caused by mutations on a single locus (genomic coordinate) and their inheritance follows Mendel's laws. Therefore, they are relatively easy to detect and confirm. Most of them are caused by SNPs for which one allele is disease associated and the other isn't. The best known example of a Mendelian disease is sickle-cell anemia.[33] Detecting them is usually done by PCR or SNP array. SNP array is the DNA microarray that uses DNA hybridization and fluorescence microscopy to detect polymorphisms in a sample of genetic material.

Large-scale sequencing projects uncovered many SNPs in the human genome. At the same time, advancements of chip technology enabled multiplexing of the DNA hybridization

procedure through the ever-decreasing size required for of a single hybridization event. This has enabled relatively cheap assessment of large number of SNPs and launched a new area of research called Genome Wide Association Studies (GWAS). These studies try to uncover another level of genomic influence on phenotype. Mendelian diseases and, in general, phenotypic traits that are affected by a single locus are rare. Most diseases and traits have a complicated network of underlying interactions and many loci of low to moderate effect contribute to their signal. GWAS studies try to analyze these and find SNPs that are associated with different phenotypic traits. Many new insights into the "nature" part of our phenotype were gained through these studies, but some expected associations were either not detected, or had insufficient explanatory capacity. A reason for this is that GWAS looks at only one type of genomic variation. Although SNPs are the most numerous of these, and span the entire genome, overall, more diversity (in terms of the number of different nucleotides) is achieved through other structural variants.[31]

To uncover the effect that larger variants have on the phenotype, DNA microarrays are not sufficient. Instead, the whole-genome sequencing is required. Current sequencing technologies have only recently advanced to the stage where enough individuals can be sequenced at appropriate depth to make association studies feasible. This type of whole-genome analysis would enable associating larger structural variants with phenotypic traits in a similar way that GWAS did with SNPs. Additional problem is the read length that current sequencing technologies produce (see chepter 2.3.) Whole-genome sequencing requires mapping of the obtained reads onto the reference sequence, which reduces the bioinformatical capacity to detect larger structural variants. Complete *de novo* assembly of each sequenced genome is the only way to capture our full genomic diversity and achieving this would enable the final stage in associating genomic variation to phenotypic traits and diseases.[34]

2.3. Sequencing methods

The first widely used DNA sequencing method is accredit to Frederick Sanger.[35] It modified the existing primer-extension strategy[36] enabling more rapid DNA sequence determination and thus making it applicable to larger genomic regions. In 1977, first fully sequenced

genome, 172,282 long bacteriophage ϕX174 nucleotide sequence was reported.[37] Many more genomes were sequenced after this and the use of Sanger sequencing method expanded, earning him his second Noble Prize for Chemistry in 1980. This process culminated with the Human Genome Project (HGP), world's largest collaborative life science project. It was launched in 1990 with the $3 billion grant from the US government and ambitious objective of determining complete base pair sequence that makes up human DNA as well as identifying and mapping all of the genes from both physical and functional standpoint. This process was marked with several controversies, most notably, separation of a part of the HGP consortium, led by Craig Venter, due to disagreement over the method that was to be used in the project. The result was a simultaneous, privately funded project, with the same objective, and led by Venter's company, Celera. They utilized modification of the original Sanger sequencing method called Shotgun sequencing.[38] Additional tension was raised between public and private sector after disagreement about intellectual property protection and legal status of genes and genomic regions. However, close to completion of the initial sequencing, the two groups united and in 2001 together published the first draft of the human genome.[39] Several improved versions followed with each of them containing fewer gaps in the sequence. Reference human genome is still incomplete. However, there are only ~100 gaps corresponding to <0.01 % of the overall sequence.[40] HGP is still active, working towards its initial goal, and expending it. In June 2016, HGP-write, ambitious 10 year project extension was announced with the objective of synthesizing the three billion nucleotide long human genome.[41]

Determining genetic sequence triggered unprecedented scientific and medical advancement, which in turn led to research and improvement of the sequencing methods. Although powerful, Sanger sequencing had several shortcomings. Most notably, high per base sequencing cost and time-consuming preparation procedure. In the following years, several new methods were developed. Due to their matching objectives, they are collectively dubbed Next-generation sequencing (NGS) methods[42] (table 1). Few most important methods will now be described in detail.

Table 1: Some sequencing methods.

Method	Read length / bp	Accuracy (single read not consensus)	Time per run	Cost per 1 million bases / US$
Single-molecule real-time sequencing (Pacific Biosciences)	10,000 to 15,000	87% single-read accuracy	30 minutes to 4 hours	$0.13–$0.60
Ion semiconductor (Ion Torrent sequencing)	up to 400	98%	2 hours	$1
Sequencing by synthesis (Illumina)	varies between instruments, usually 50 to 300	99.9% (Phred30)	1 to 11 days	$0.05 to $0.15
Chain termination (Sanger sequencing)	400 to 900	99.9%	20 minutes to 3 hours	$2400
Nanopore sequencing	up to 100,000	currently low	flexible, as low as 30 minutes	very low

2.3.1. Next Generation sequencing

Pyrosequencing is a DNA sequencing technique that relies on detection of pyrophosphate ($P_2O_7^{4-}$, PP_i) released during nucleotide incorporation.[43] It is a type of "sequencing by synthesis" technique, which means that the nucleotide content is determined while the in vitro DNA replication process is performed. In the case of pyrosequencing, this is achieved by monitoring pyrophosphate release that accompanies nucleotide incorporation in the growing DNA chain.

The process begins by template preparation for which emulsion PCR (emPCR) amplification is used (figure 2.7a). A library of target DNA regions (fragmented or mate-paired) is prepared and adaptors containing universal sequences are ligated to their ends. These universal primers enable the amplification of all genetic material with common PCR primers. After ligation, DNA is separated into single strands and captured onto beads under conditions that favor one DNA molecule per bead. Once DNA is captured, emPCR amplification is performed, resulting in multiple copies of DNA fragment from targeted genomic region on each bead. These are deposited into individual PicoTiterPlate wells[44] in which the sequencing is performed. Large number (1-2 million) of wells available for this process enables multiplexing of the method. Pyrosequencing is in its essence a bioluminescence method. The release of pyrophosphate during nucleotide incorporation is converted into visible light using series of enzymatic reactions (figure 2.7b). Different nucleotides, in form of dNTPs are added consecutively to the reaction. If the added nucleotide is complementary to the next un-paired nucleotide in the fragment on the bead, DNA polymerase will incorporate it and thus extend the strand. Simultaneously,

pyrophosphate is released. To convert this into measurable signal, separate beads that have ATP sulfurylase and luciferase attached to them are loaded into wells surrounding the template beads. Additionally, wells are incubated with adenosine-5-phosphosulfate (APS) and luciferin. This produces a series of reactions that end with the light signal:

$$P_2O_7^{4-} + APS \Leftrightarrow ATP + SO_4^{2-}$$

$$ATP + luciferin \Leftrightarrow luciferyl\ adenylate + P_2O_7^{4-}$$

$$luciferyl\ adenylate + O_2 \Rightarrow oxyluciferin + AMP + \boldsymbol{hv}$$

Therefore, only wells that contain the bead, which has the nucleotide complementary to the one that is being incorporated, will produce this bioluminescence signal. Light is measured with the charge-coupled device camera and its intensity is proportional to the amount of pyrophosphate available to initiate reactions. Therefore, if multiple nucleotides are incorporated (due to consecutive nucleotides in the DNA fragment), more pyrophosphate will be released and the higher intensity measured (figure 2.7c). This effect imposes accuracy limitations to the method since after certain number of consecutive nucleotides, light intensity differences become indistinguishable, and the precise number of nucleotides in the DNA fragment can't be determined.

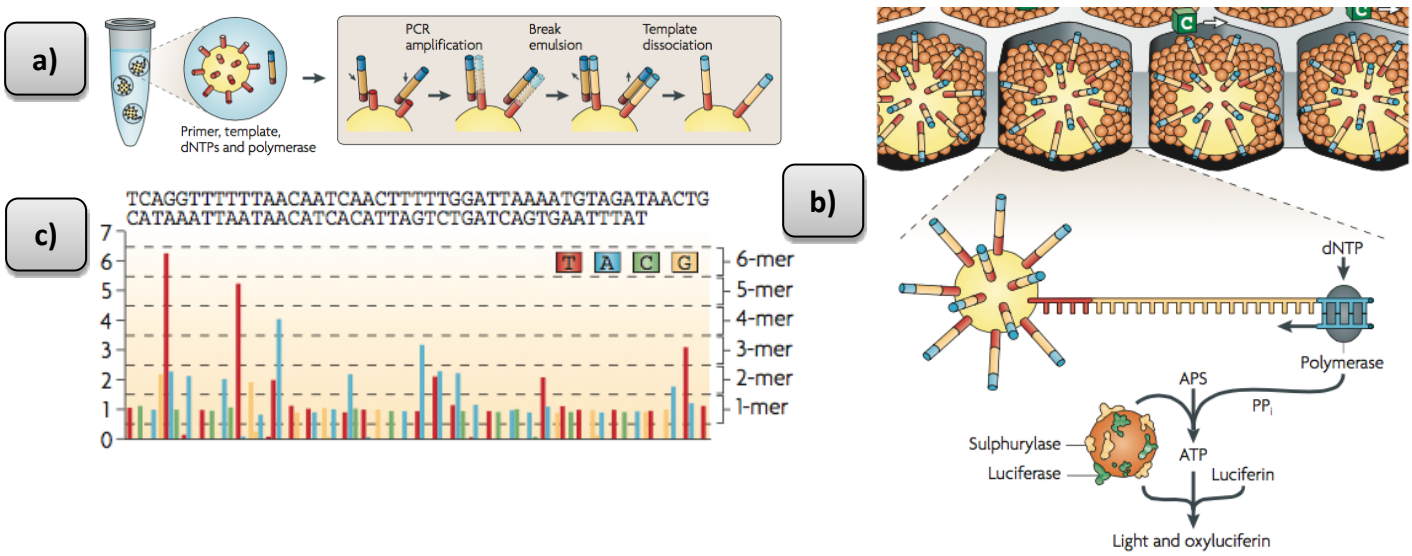


Figure 2.7: Pyrosequencing procedure, **a)** emPCR, **b)** converting pyrophosphate into the light signal by a series of enzymatic reaction and **c)** flowgram.[42]

Another “sequencing by synthesis” platform was developed by the Illumina Company. Their sequencing method is based on modified deoxyribonucleotides that can act as reversible terminators during DNA synthesis.[45] This technology is currently dominating the sequencing market. Illumina uses solid-phase amplification to produce copies of targeted DNA fragments. Fragments are again ligated with the adaptor sequence and, in this method, primed to the complementary sequences attached to a glass slide (figure 2.8a). After that, bridge amplification is performed where each fragment is connected to the adjacent immobilized primer and copied by the DNA polymerase. This forms clusters of fragments, which increases the base calling sensitivity. Sequencing proceeds through sequential incorporation of modified nucleotides (figure 2.8b). First modifications they contain are terminating groups that prevent polymerase activity. Second modifications are fluorescent dyes. Each nucleotide is labeled with a different dye and can therefore be distinguished by the fluorescent signal. In each sequencing step, all modified nucleotides are added into the reaction mixture. Mutated DNA polymerase is used for the incorporation of modified nucleotides and in every step only one, complementary to the next un-paired nucleotide on the DNA fragment, is added to each strand belayed on the glass slide (figure 2.8c). Synthesis can’t proceed until the terminating group is removed, but before this, unused nucleotides are washed away and fluorescent signals measured. This way, each step corresponds to the extension of targeted DNA fragments by one position and all four nucleotides are examined simultaneously. Output of this method is a nucleotide sequence of each DNA fragment cluster scattered on the glass slide (figure 2.8d).

Single-Molecule Real Time (SMRT) sequencing utilizes somewhat different sequencing approach. As the name suggests, it attempts to determine genetic sequence in real time and from a single DNA polymerase molecule.[46] To this end, sequencing is done on a chip that contains many small wells (figure 2.9a). At the bottom of each well, a single stranded DNA fragment is immobilized with an active DNA polymerase molecule synthesizing its complementary strand (figure 2.9b). Nucleotides that are being added are modified, each with a different fluorescent label, which produces light signal at the time of their incorporation. Genetic sequence is determined from the continuous measurement of fluorescence spectra. Unlike other methods, SMRT doesn’t use any signal amplification procedure. It therefore requires very sensitive measurement device. To this end, zero-mode waveguide (ZMW) detectors are utilized.[47] This is the main technological advancement that

enabled the SMRT procedure. The advantage it offers is that amplification step can be omitted. Sequencing is done in real time and whole process is a lot faster as it doesn't require any reagent exchange or termination stops that are part of other NGS procedures. Problem with this method is that single DNA molecule detection causes base calling accuracy problems since there are no averaging effects that enhance signal-to-noise ratio. Combined with increased polymerase error rate, due to fluorescent modifications on the nucleotides, this causes a significantly lower read quality.[48] However, some modifications have already been purposed that improve the base calling accuracy.[46]

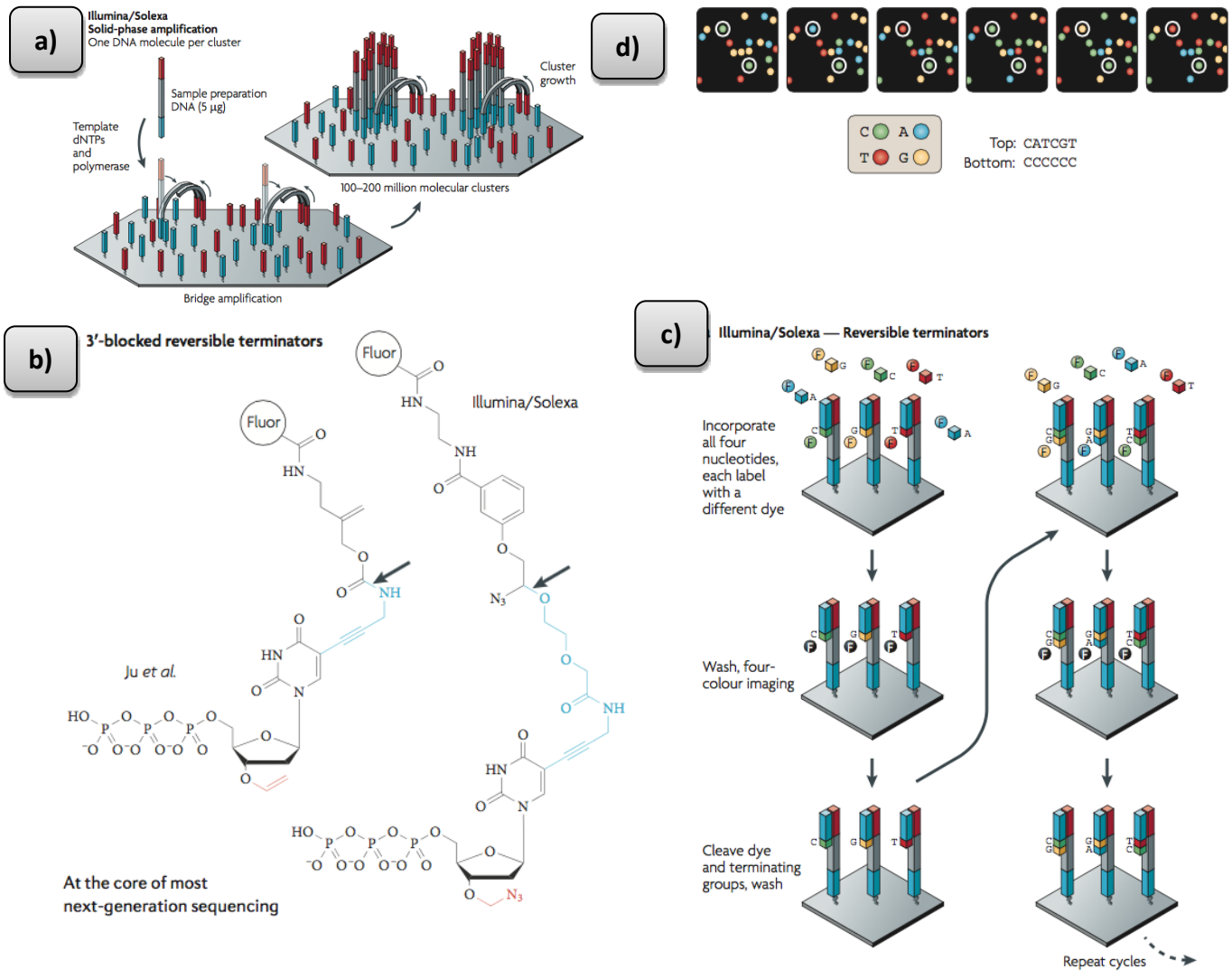


Figure 2.8: Illumina sequencing, **a)** bridge amplification on a glass slide, **b)** modified nucleotides, **c)** sequencing procedure, **d)** output of this method.[42]

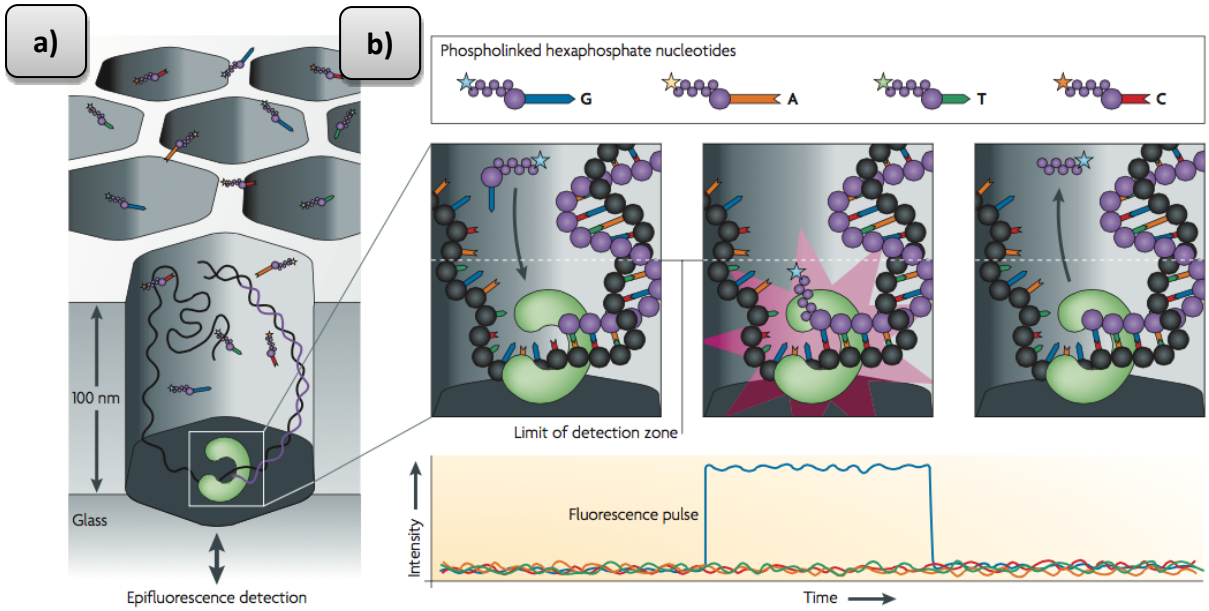


Figure 2.9: SMRT sequencing, **a)** chip layout, **b)** sequencing procedure.[42]

2.3.2. 3rd generation sequencing and nanopore

There are several new methods in development that try to simplify the sequencing process and avoid iterative procedure that most NGS technologies use. NGS has a lot of improvements over Sanger sequencing, especially in terms of time and cost reduction. However, some aspects haven't been sufficiently advanced. Most notably, read lengths, which remain the bottleneck for many sequencing applications. Reads produced by these methods are relatively short and complicated bioinformatical algorithms required to extract information from them (see chapter 3.3.) This reduces variant discovery capabilities of NGS as larger variants often pass unnoticed. New methods, informally dubbed the third generation, try to address this problem as well as further reduce the sequencing cost and simplify the procedure.[48] Some of these methods in development will be briefly discussed.

Transmission electron microscopy approach tries to detect atoms which uniquely identify individual nucleotides.[49] This method was envisioned by Richard Feynman even before Sanger sequencing was developed.[50] Electron microscopy can obtain resolution of up to 100 pm which is sufficient to observe small biomolecules and DNA can be seen easily (figure 2.10). However, this resolution isn't sufficient to decipher individual nucleotides. Therefore, the DNA sample has to be selectively labeled with heavy atoms which is still a significant technological challenge.[51] Another sequencing method in development

attempts to utilize tunneling currents to differentiate between nucleotides.[52] This method would operate on single DNA molecules confined to the conductive surface and read genetic code using scanning tunneling microscopy. This type of sequencing would be orders of magnitude faster than any other purposed or developed method, but there are significant technological barriers that still need to be overcome. Most notably, no methods have been described that would enable stretching and confining any longer DNA fragment to the conducive surface which is required for this sequencing process.

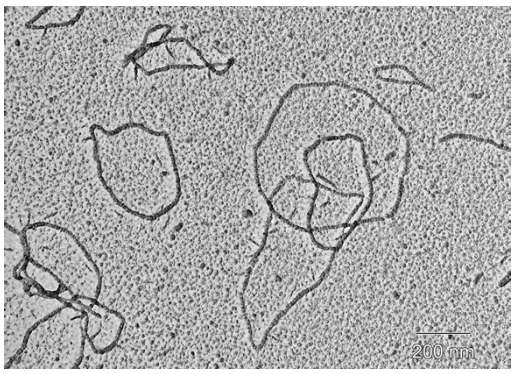


Figure 2.10: Negative Stain TEM of DNA-RecA Complex.[53]

Nanopore sequencing, developed by Oxford Nanopore Technologies, is currently the only commercially available third generation method.[54] Theoretical premise of the method is that the steady stream of ionic current (electric current due to the conduction of ions) through the pore of nanoscale dimensions gets modified when single stranded nucleic acid passes through it. If this change is characteristic for each nucleotide, then the genetic sequence can be deciphered in this way (figure 2.11a). Method is in development for almost 30 years and was commercialized recently with the release of the MinIon device[55] (figure 2.11b). The essence of this technology is an engineered protein nanopore, similar to the transmembrane proteins found in living cells. Nanopore is incorporated into an electrically resistant polymer membrane. Across the membrane voltage is applied causing the ionic current across the pore, which gets disrupted if an analyte passes through it. This enables base calling of single stranded nucleic acid molecules. DNA strands are separated by the helicase enzyme that was engineered to specifically recognize the nanopore protein. The enzyme also feeds the chain through the pore. As the chain moves, characteristic disruptions to the ionic current are recorded and translated into the nucleotide sequence.

Nanopore technology has great potential, primarily due to the simple sequencing method at its core which doesn't require any time-consuming preparation steps and can produce very long reads.[56] An additional advantage is a wide variety of available application environments, some of which are inaccessible to traditional sequencing methods.[57] Small device size also offers a distinct scaling opportunity which is already being exploited with Promethion, a bench-top sequencing system under development (figure 2.11c). However, a lot of challenges also remain. The most significant is the error rate, which is still very high. This is due to low signal-to-noise ratio of the ionic current changes for different nucleotides as well as the size of the system. The nanopore protein is much larger than individual nucleotides that pass through it and many nucleotides affect the ionic current at all times which is why complex algorithms are required to discern individual effects. A lot of effort is invested into tackling this problem, both by continuous modification of protein nanopores, and base calling algorithm optimizations, and a steady improvement of the sequencing accuracy is observed.[58]

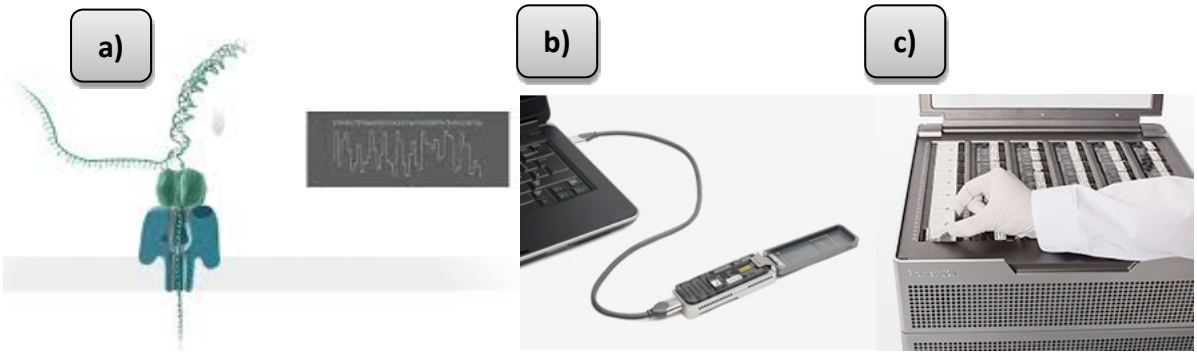


Figure 2.11: Nanopore, **a)** sequencing scheme, **b)** MinIon, **c)** PromethIon.[54]

2.4. 1000 Genomes Project

The 1000 Genomes Project was an international research effort aimed at establishing detailed catalogue of human genetic variation. Initial plan, made in 2008 when the project was launched, was to sequence the genome of at least one thousand anonymous participants that span several ethnic groups. Final data set, published in 2015 when the project was completed, contained data for 2504 individuals from 26 populations (figure 2.12).



Figure 2.12: Populations in the 1000 Genomes Project.[59]

Motivation for the 1000 Genomes Project came from various studies that expended the results of initial human genome sequencing, revealing prominent genetic diversity between and within populations.[60] Furthermore, various structural variants were shown to exert influence on phenotype and contribute to genetic disease background. By far the most prominent of these, by measure of occurrence frequency, are SNPs, which were the topic of numerous GWAS that revealed loci involved in a wide range of traits - from human height[61] to disease susceptibility[62]. Another level of genetic diversity is caused by copy-number variants (CNVs), which are still used as genetic markers for forensic identification.[63] All of this indicated necessity for a rigorous and all-encompassing human genetic variation analysis.

The 1000 Genomes Project was divided into 4 phases. Results from the initial, pilot phase were published in 2010.[64] This consisted of low coverage whole-genome sequencing from 180 samples, high coverage whole-genome sequencing for 2 mother-fathter-child trios and high coverage sequencing of 1000 gene regions in 900 samples. Main purpose of this phase was to establish the sequencing depth required to obtain data of sufficient quality as well as assessing strategies for data sharing across samples, and was followed by 3 main project phases. Each of these consisted of low coverage genome and exome sequencing of all samples as well as high coverage genome sequencing for small number of individuals used for validation purposes. The number of individuals was gradually increased in each

phase. Final results from main phase 3 were published in October 2015[31][65] and were comprised of 2504 individual genomes, 24 of which included high coverage genome sequence.

Final results revealed over 88 million variants, 84.7 million of which are SNPs. A typical genome differs from the reference human genome at 4.1 to 5.0 million sites. Most of these sites (>99 %) correspond to SNPs. However, other structural variants affect more bases in the genome sequence overall. The total number of observed non-reference sites differs greatly among populations and is the highest in samples from African ancestry (figure 2.13). This is consistent with out-of-Africa model of human origins that predicts longest variant accumulation time for African populations. The majority of variants are rare. ~64 million have frequency <0.5 %, ~12 million have frequency between 0.5 % and 5 % and ~8 million >5 %. However, most variants in a single genome correspond to high frequency ones. Only 1 - 4 % of those variants have frequency <0.5 %. With respect to the protein-coding regions, a typical genome was found to contain 149 - 182 sites that introduce stop codon and thus cause protein truncating variants, while 10,000 to 12,000 sites correspond to peptide-sequence-altering variants. African Genomes were consistently at the high end of these ranges as well. This pattern wasn't observed in respect to abundance of disease causing variants. There were 24 - 30 variants per genome implicated in rare diseases, with European ancestry individuals at high-end of these counts. This is most likely due to the ethnic bias of current genetic studies.[31]

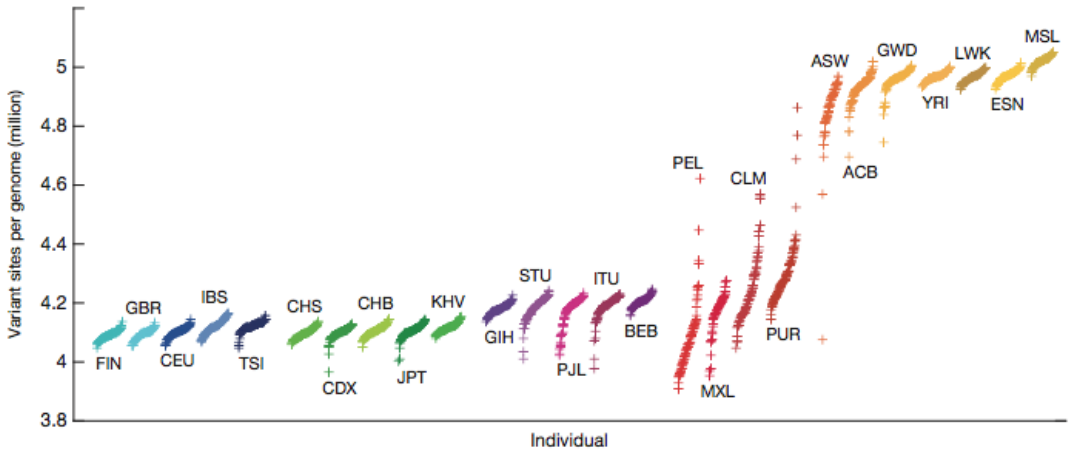


Figure 2.13: Distributions of the number of variant sites in individual genomes for populations sequenced in the 1000 Genomes Project.[31]

As already mentioned, structural variants other than SNPs, although less numerous, affect more positions in the genetic sequence. 1000 Genomes Project also reported an integrated set of eight structural variant (SV) classes. Emphasis was put on major classes of SVs, defined as those that affect ≥ 50 bp. Overall, 68,818 SVs with average size of 5.57 kbp were identified.[65] SNPs will be the only variants considered in this thesis since only their occurrences in the protein-coding regions of the genome can be directly translated into the amino-acid substitutions.*

* In this case the term "variant" is used in broader sense since, by previous definition of SVs, SNPs are not in this category.

§ 3. Theoretical background

3.1. Probability distributions

Probability distribution is a mathematical descriptions of a random phenomenon in terms of the probabilities of events. Events are the set of all possible outcomes of the phenomenon being observed. This set is called the sample space. For example, the sample space of a coin flip is: {heads, tails}, since this represents all possible outcomes of a phenomenon being observed.

There are two classes of probability distributions: discrete and continuous. Discrete probability distribution describes a probability of each outcome. Generally, it's used when there is a finite number of possible outcomes. The coin flip is an example of a discrete phenomenon. List of the probabilities of all outcomes for a discrete probability distribution is given by the probability mass function (PMF). Another simple example is the roll of a fair, six-sided dice. In that case there are 6 possible outcomes and the PMF is a function that assigns the probability of 1/6 to each dice value (1 to 6). Continuous probability distribution can be described by the probability density function (PDF). The idea is analogous to the discrete case, but the properties of a PDF are more complex.

A PDF of a continuous random variable is a function that describes the relative likelihood for this random variable to take on a given value. The probability of it falling within a particular range of values is given by the integral of this variable's density over that range. The probability density function must be nonnegative everywhere, and its integral over the entire space equal to one. The probability of any individual outcome is 0.

A probability distribution whose sample space is the set of real numbers is called univariate, while a distribution whose sample space is a vector space is called multivariate. Only univariate distributions will be considered hereafter.

Another closely related concept is the cumulative distribution function (CDF). CDF of a real-valued random variable X, evaluated at x, is the probability that X will take a value less than or equal to x. Precise mathematical definitions are not necessary to understand the use of these functions and therefore won't be discussed, but it should be noted that in probability theory, defining either CDF of PMF are valid ways of specifying a discrete

probability distribution, while a continuous one can be specified by supplying either its PDF or CDF (there are also other ways for defining both of these).

The idea of probability distributions underlies the mathematical disciplines of probability theory and statistics. As such, it is used in any scientific field where the probability and statistics themselves are relevant. The basic understanding of concepts introduced in these disciplines and their application are therefore essential in any scientific research, since they are intrinsic to all systems, from elementary particles to biological populations. These concepts will now be explored for some frequently used probability distributions.

3.1.1. Continuous uniform distribution

The simplest continuous probability distribution is the uniform, also called a rectangular distribution. This is a family of symmetric probability distributions such that for each member of the family, all intervals of the same length are equally probable (figure 3.1). Different members of the family are defined by the arbitrary values of parameters *a* and *b* in the PDF of the distribution:

$$f(x) = \begin{cases} \dfrac{1}{b-a} & \text{for a} \leq \text{x} \leq \text{b} \\ 0 & \text{otherwise} \end{cases}$$

If *a = 0* and *b = 1*, the resulting distribution is called a standard uniform and denoted *U(0,1)*.

In statistics, when a p-value is used as a test statistic for a simple null hypothesis, and the distribution of the test statistic is continuous, then the p-value is distributed according to *U(0,1)* if the null hypothesis is true.

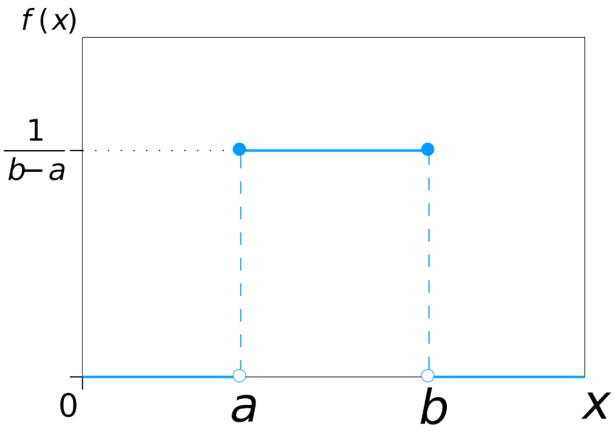


Figure 3.1: Continuous uniform distribution.

3.1.2. Normal distribution

The normal distribution is a very common continuous probability distribution that is often used in life sciences to represent random variables whose distributions are unknown (figure 3.2). The PDF of this distribution is:

$$f(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\sigma^2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

where μ is mean of the distribution and σ its standard deviation. These two parameters define the family of normal distributions and the special case in which $\mu = 0$ and $\sigma = 1$ is called the standard normal.

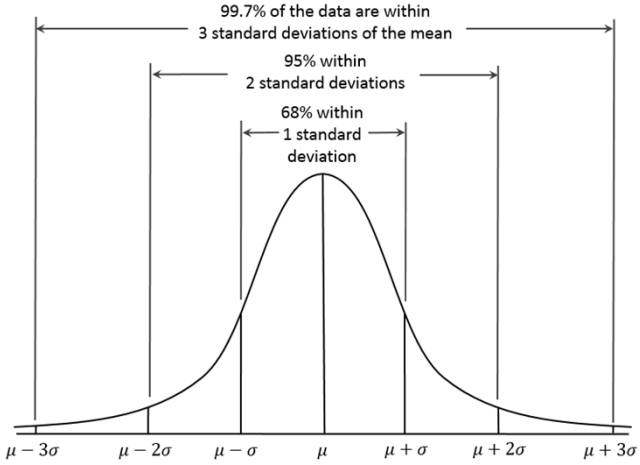


Figure 3.2: Normal distribution.

The most important use for this family of distributions comes from the central limit theorem. This mathematical theorem states that averages of random variables independently drawn from some unknown distribution converge in their distribution to normal, that is, become normally distributed when the number of random variables is sufficiently large. What this means is that even if we don't know the precise distribution of our variable, given the large enough number of observations, averages of these observations will be normally distributed (if we repeat the sampling many times, averages of these samples will be normally distributed with some mean and standard deviation). This theorem is valid under some mild mathematical restrictions on the underlying distribution. If the sampling is done from distribution with mean μ and variance σ^2, the derived normal

will have the same mean (μ) and variance $\frac{\sigma^2}{n}$, where n is the number of observations in each average.

For example, sampling from previously discussed continuous uniform distribution produces the distribution of averages that increasingly resembles normal with larger number of observations (figure 3.3a). This property is even more obvious if sampling is done from some unknown distribution, which is evidently far from normal (figure 3.3b).

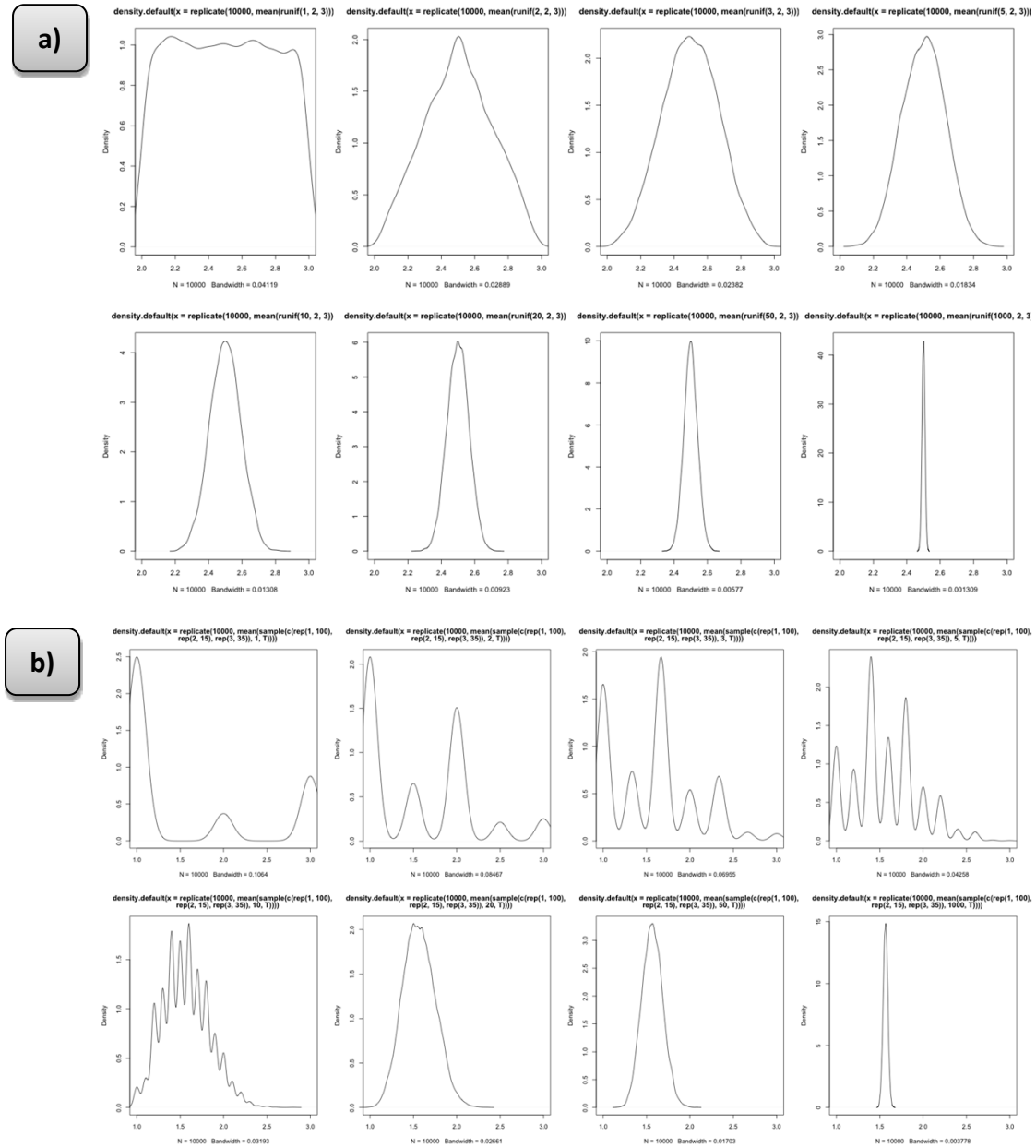


Figure 3.3: Plots of average values for 1000 samples with increasing number of observations in each sample (1, 2, 3, 5, 10, 20, 50 and 1000) from **a)** continuous uniform distribution and **b)** some random distribution.

Another use of the normal distribution is for modeling measurement errors. If all variables of the measured property have been taken into account, than the errors observed in a physical experiment should be distributed according to the normal distribution. Any divergence from this behavior can be used as an indication that the model of the experiment needs to be changed.

3.1.3. Poisson distribution

Contrary to the previously discussed examples, Poisson distribution is a type of the discrete probability distribution. It defines the probability for a certain number of events occurring in a fixed interval of time, given that these events occur with a known average rate and independently of one another. Some examples of this behaviour are the number of decay events from a radioactive source occurring each second (or minute, hour, *etc.)* or the number of mutations introduced to a fixed length of DNA strand. The only parameter of the distribution, λ, is called the event rate and tells the average number of events in the given interval (figure 3.4). Poisson PMF is:

$$P(k\ events) = \frac{\lambda^k e^{-\lambda}}{k!}$$

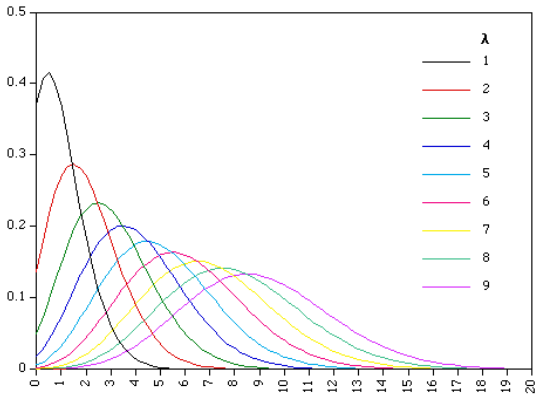


Figure 3.4: Poisson distribution for different values of parameter λ.

This distribution is used in many biological problems because it's a robust predictor of occurrence frequencies for various events. In biochemistry, the number of cells (or cell colonies) that are successfully transformed with the foreign plasmid in a given time interval will follow the Poisson distribution. If we know the event rate, we can predict the number of

cells that are transformed and use this information to decide how many we will submit to further testing.

3.1.4. χ^2 distribution

The χ^2 distribution with k degrees of freedom is the distribution of a sum of squares from k independent standard normal random variables. It is one of the most widely used probability distributions in inferential statistics.[66] It's also the most extensively used distribution in this thesis.

In mathematical notation, if $Z_1,\ldots,Z_k$ are independent, standard normal random variables, than the sum of their squares:

$$Q = \Sigma_{i=1}^{k} Z_i^2$$

is distributed according to the χ^2 distribution with k degrees of freedom (figure 3.5). This is denoted as $Q \sim \chi^2(k)$ or $Q \sim \chi_k^2$. Number of degrees of freedom, k, is the only parameter of the distribution. Its PDF is:

$$f(x) = \frac{1}{2^{k/2}\Gamma(\frac{k}{2})} x^{\frac{k}{2}-1} e^{-\frac{x}{2}}$$

From this equation it can be seen that the χ^2 is a special case of the more general Γ distribution.

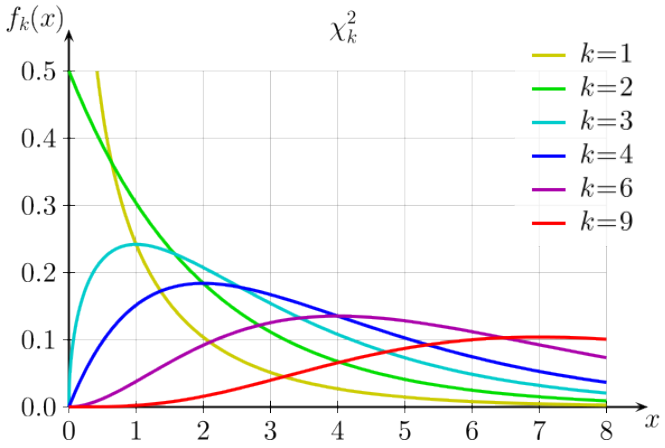


Figure 3.5: χ^2 distribution for different values of parameter k.

This distribution is used in a wide variety of situations. The most common application is for two types of hypothesis tests: independence in contingency tables and goodness of fit

of observed data to a hypothetical distribution. It turns out that both of these problems follow the above described χ^2 distribution. Mathematical proof for this won't be shown, but the relationship between tests and specific distributions will be explored later (see chapter 3.2.) For contingency table independence, parameter $k = N - 1$, where *N* is the dimension of the table. It is the same for goodness of fit test, only there *N* represents the number of classes into which data is divided. Since they follow the χ^2 distribution, tests themselves have been named χ^2 contingency and χ^2 goodness-of-fit test.

In biological terms, any problem that requires one of these tests will use χ^2 distribution to assess the significance of the observed effect. Common example for the contingency test is measuring differences of effect between males and females or between multiple specimens. Goodness-of-fit test can be used to assess whether some observations (*e.g.* medical) significantly deviate from their predicted values and it will be extensively used for this purpose.

Apart from the hypothesis testing, χ^2 is used in many other problems of statistical inference. Due to its role in Student's t-distribution, it is involved in estimating the mean of a normally distributed population or the slope of a linear regression. Additionally, it's important for any variance analysis problem in which F-distribution is used, since F-distribution itself is just the ratio of two independent χ^2 random variables.

3.2. Statistical hypothesis testing

Hypothesis testing is the formal procedure in statistics used to accept or reject statistical hypotheses. The procedure consists of proposing the statistical relationship between some datasets and comparing it to an idealized null hypothesis that no relationship exists. Most often, two datasets are compared. They can either both be obtained by some sampling method, or one can be obtained that way and the other constructed from an idealized model based on the null hypothesis. The comparison is deemed statistically significant if the observed relationship between datasets would be unlikely given that the null hypothesis is true. The term "unlikely" is precisely defined by the threshold probability, which is chosen

independently in each situation and depends on the relationship that is being assessed and the precision required.

3.2.1. Null hypothesis vs. alternative hypothesis

Stating the null and the alternative hypothesis, as described above, is the crucial step of the testing procedure. These two hypotheses must be composite and mutually exclusive, meaning that they have to cover the whole space of possible testing outcomes and that they can't both be true at the same time. Failure to do this will blur the interpretation of the results because the question that needs to be answered won't be the direct output of the statistical test. This is called "garbage in, garbage out" (GIGO) principle.

The null hypothesis (usually denoted H_0) will always assert that any discrepancy in the data results purely from chance. If we test two samples of some physical quantity, null hypothesis will claim that their average values are the same. If we test the fairness of a coin, the null hypothesis will claim that any discrepancy between the number of heads and tails results from randomness of a fair coin. In the criminal trial, the null hypothesis will be that the defendant is not guilty, that is, that any apparent evidence against him/her occurs accidentally.

Alternative hypothesis (usually denoted H_1) asserts that the observed data is influenced by some non-random cause, that is, that there are additional variables that influence the data which have not been taken into account by the null hypothesis. In the previous examples, the alternative hypotheses would claim that in two sets of observations, there is additional effect on the observed physical quantity, which differs between datasets. Therefore their averages would, under this hypothesis, differ as well. The alternative hypothesis for coin flips would claim that the coin is unfair, that is, that discrepancy between the number of heads and tails is caused by coin's preference for one of two sides. In the criminal trial, the alternative hypothesis is always that defendant is guilty and that the evidence against him/her are truthful.

The null hypothesis is always assumed to be correct prior to the statistical test and every test only tries to reject it. Observations are first analyzed under the scope of the null hypothesis. Then, the discrepancy in the data is quantified in terms of the p-value, which gives the probability that the observations indeed result from this null hypothesis. Finally, if

the p-value is lower than some defined threshold, the null hypothesis is rejected and the alternative hypothesis accepted. If p-value is not lower than the threshold, the null hypothesis is not rejected.*

Last thing to mention here is that hypotheses can be stated to allow for a one-sided (one-tailed) or a two-sided (two-tailed) test. In one-sided case, the alternative hypothesis might claim that some value is strictly smaller than expected (or it might claim that it's strictly larger), while in two-sided case, it will simply claim that some discrepancy exists and won't specify the direction. From previous examples, one-sided tests would be that the average value of a physical quantity from one sample is larger than from the other (in that case the null hypothesis would be that the difference between the two averages is ≤ 0) and that the coin prefers heads (in which case the null hypothesis would be that the coin is fair or that it prefers tails). Which type of test is used depends on the data that we have and the question we want to answer.

3.2.2. Test statistics and p-value

The decision whether to reject the null hypothesis of any statistical test is based on the p-value. The p-value states the probability that the observed data stems from the null hypothesis. Small p-value therefore indicates the alternative hypothesis to be true and large p-value indicates the null. The largest part of the statistical testing procedure is obtaining this quantity. There are several ways to do this. The most frequently used method, which will be explained further, is calculating the p-value from some test statistic. Some of the alternative methods are calculating it from repetitive simulations, estimating it from results of the analysis (the most uncertain method), or obtaining it through visualization of the appropriate data plots.[67]

Test statistic is a numerical summary that reduces the data to a single value and satisfies the following condition: sampling distribution of the test statistic under the null hypothesis must be calculable, either exactly or approximately. This condition ensures that the p-value can be directly calculated from the statistic. In practice, this means that the

* It's important to note the terminology here. Due to the setup of the testing procedure in which the null hypothesis is assumed to be correct, failure to reject the null hypothesis (p-value above the threshold) shouldn't be called "accepting the null hypothesis", since this is already established at that point and such claim would imply that it was established only now.

method we choose to reduce our data and calculate the statistic must give a result that follows a valid probability distribution. Satisfying this condition allows us to directly correlate the value of the statistic with a p-value. Choosing the appropriate test statistic is often the hardest part of hypothesis testing since we have to *a priori* know the distribution it will follow. Choosing an inappropriate test statistic, which is not in accordance with proposed hypotheses will make the test result (or more precisely, our interpretation of it) invalid, as suggested by the GIGO principle. There are however many tests available, some of which are applicable in wide variety of situations. Examples of these are nonparametric tests like Kolmogorov-Smirnov, or Wilcoxon rank sum test, and it's often a good idea to use these when in doubt of the distribution that the data follows.

Proceeding with our previous examples, for observations of a physical quantity we can test whether the two observed samples come from same distribution by applying Student's t-test. That gives us test statistic which follows t-distribution and from it we can directly determine p-value. If we are not sure whether our samples are normally distributed, which is a requirement for the t-test, we can instead use nonparametric Wilcoxon rank sum test to see whether two data samples have the same mean, regardless of their underlying distributions. This produces the test statistic W, which follows a complicated probability distribution, but still directly correlates to a p-value. The coin flip example is simpler, our test statistic for that case can simply be the number of heads (or tails) in a given number of coin flips. That value is known to follow a binomial distribution and therefore directly relates to the p-value. The court trial example is more complicated since it can't be resolved by any test statistic. Therefore, other methods for estimating the p-value (and subsequently, the defendant’s guilt) are required.

3.2.3. Decision Errors

The described hypothesis testing procedure ends with the p-value from which the decision whether to reject the null hypothesis is made, based on the chosen threshold probability. In this step the probabilistic result (p-value) is converted into a binary (either reject or don't reject) and this contains an intrinsic error risk.

In every test, there are two possible errors we can make: we can reject the null hypothesis when in reality it is correct or we can fail to reject it when in reality it is wrong.

These are called type I and type II error, respectively (figure 3.6). Both of these can always be made, regardless of the p-value that was obtained in the test. However, their frequency can be controlled with the threshold probability. Since we decide which threshold to use, we can *a priori* know their relative probabilities.

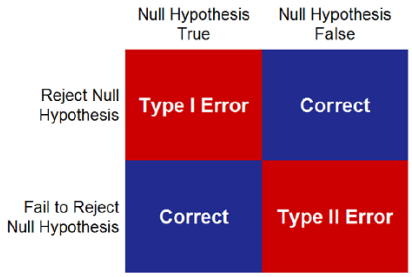


Figure 3.6: Hypothesis testing errors.

In type I error we claim that there is additional effect in the observed sample, which hasn't been taken into account by the null hypothesis. However, no such effect exists. That's why this error is also called a false positive. The probability of committing a Type I error is called the significance level and is often denoted by α. For a specific statistical test in which the null hypothesis is rejected, this probability is simply equal to the observed p-value.

In type II error we claim that the observed sample is described by the null hypothesis when in reality, there are additional factors. That's why this error is also called a false negative. The probability of committing a type II error is often denoted by β and the probability of not committing it ($1 - \beta$) is called the statistical power of the test. The power can also be defined as the probability of correctly rejecting the null hypothesis (the two definitions are equivalent). Estimating the power for a specific test in which the null hypothesis was rejected is complicated and it may depend on a number of factors some of which are the probability threshold, magnitude of the effect of interest in the population, used sample size *etc*.

It's important to note that the rates of type I and type II errors are inversely proportional and controlled by the applied threshold. If we use high threshold probability, we will relatively often reject the null hypothesis, which makes the type II error unlikely, but increases the probability of the type I error. Adversely, low threshold probability will ensure a small number of false positives, but will also increase the probability for a false negative.

Which situation is better depends on the problem that we are testing. If we are analyzing a set of genes as an initial filtering step for our research, we'll use high threshold probability to ensure we collect all potential candidates and won't mind if some of them don't end up being significant. For the court trial example, we'll use a very conservative threshold that will minimize the number of false positives (wrongful convictions), although we have to keep in mind that this will also increase the probability for a false negative (acquittal of the guilty).

3.3. Bioinformatics

Bioinformatics is a broad, interdisciplinary field that deals with *in silico* analysis of biological data. The field is being intensely developed in conjunction with improvements to the sequencing technology and it's primarily concerned with data produced by next generation methods (see chapter 2.3.1.) Despite this rapid development, vast abundance of sequencing data and the rate at which it is generated makes the bioinformatical analysis the bottleneck for many research applications.

The first step of analyzing both DNA and RNA sequencing data is read assembly. Sequencing methods generate nucleotide reads of various lengths and these need to be aligned either by *de novo* assembly or by mapping onto the reference genome.[68] Both methods involve some challenges. The mapping procedure is simpler and faster to perform, but it requires a reference genome (or transcriptome) and has problems detecting larger structural variants that significantly differ from that reference. *De novo* assembly is harder to accomplish but gives a more precise sequence as a result. For human genome and others of the similar length, mapping is the only feasible method since the reads from currently used sequencing platforms are not sufficiently long to enable *de novo* assembly.[34] New sequencing technologies are trying to address this problem (see chapter 2.3.2.) Tools used for sequence alignments are primarily developed by computer scientists and they are focused on algorithm optimization.[69]

Another major bioinformatical challenge is genome annotation, which aims to define regions of the genome that correspond to various biologically relevant features, *e.g.* genes, promotors, regulatory motifs *etc.* This process has to be automated to be employable on

larger genomes. It's usually done by looking for patterns in the genomic sequence that pinpoint to the region of interest.

Informatics has for a long time been used to complement research in evolutionary biology. Computational methods enabled tracing the evolution of a large number of organisms by measuring changes in their DNA, rather than through physiological observations. Recently, complex computational models of populations have been build and used to predict the evolutionary outcome of systems.[70]

Finally, various specialized applications of the sequencing data pose specific and often unique challenges that require new bioinformatical methods and algorithms to be developed constantly. Improvements in the research methodology enabled the application of the DNA sequencing to a wide range of biochemically interesting topics: ChIP-seq is used to find DNA regions that interact with certain proteins or histone modifications,[71] DNase-seq is used for genome-wide detections of regions sensitive to DNase I cleavage (which often correspond to segments enriched with regulatory elements)[72] and chromosome conformation capture (3C), as well as numerous modifications of this method are used to analyze the organization and interactions of chromosomes in a cell.[73] All of this suggest that application of informatics in biology will continue to expand and that bioinformatical analysis will remain a crucial component in further biological research.

3.4. Data

With the advancement of NGS technologies, per base sequencing cost has decreased dramatically, even outpacing the Moore's law (figure 3.7a). Foreseeably, the amount of genetic data available for researchers to explore has also exponentially increased (figure 3.7b). Numerous online repositories of genetic data are currently active, each focusing on different sequencing sources and methods. These won't be explored in detail and only the important aspects of few repositories relevant to the analysis will be discussed. The exact online location of individual datasets and the ways to access them are provided in the Materials and methods section.

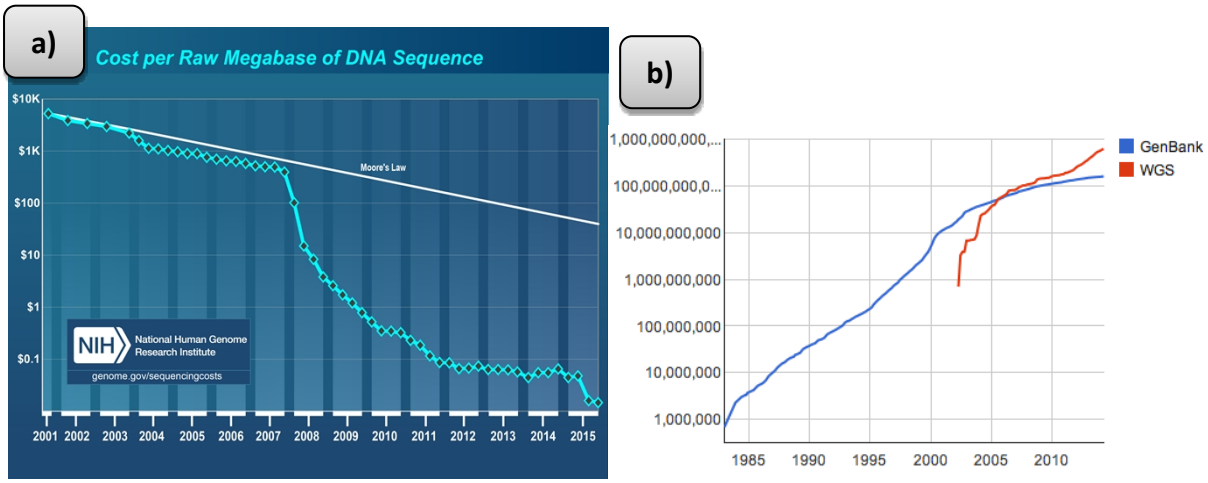


Figure 3.7: sequencing statistics, **a)** DNA sequencing cost **b)** number of nucleotide bases stored in GenBank and WGS.[74][75]

Three datasets were used in various stages of the analysis. These will be referred to as Uniprot1, Uniprot2 and 1000G and are described in detail in Materials and methods section. Every result will be accompanied with the reference to one of these datasets.

All three datasets are stored with the electronic version of this thesis on the accompanying DVD.

§ 4. Results and discussion

4.1. Map of amino acid substitutions

A map of amino acid substitutions was constructed by combining counts from individual datasets. Every entry in a dataset represents one observed amino acid substitution and is included in its corresponding position on the map. Separate maps were constructed for Uniprot1 and Uniprot2 datasets (figure 4.1.; underlying tables in extended data 1). Substitutions from Uniprot1 that can't occur due to a SNP in the reference amino acid's codon (as they require more than 1 nucleotide change) were discarded from the map (colored in white) and will be explored later (see chapter 4.5.2.) Therefore, only 150 substitutions that can occur as a consequence of a SNP are plotted.

Several things should be noted. First, counts are significantly different between two maps with Uniprot2 map having larger values. This is a consequence of different dataset sizes. It is therefore clear that the absolute values of positions don't have any direct meaning for the analysis. They do however indicate the significance of plots as they correspond to the size of underlying datasets. It's important to note that, although much smaller, Uniprot1 largely captures the overall distribution. Second, underlying datasets don't provide information on frequencies of individual variants. Therefore, although highly indicative, the relative values of different amino acid substitution counts don't entirely capture their overall distribution in a population. Third, Uniprot2 map contains one additional row and column that correspond to stop codon variants (labeled "Ter"). All of these substitutions have exceptionally small counts. And fourth, the genetic code changes required for the corresponding amino acid substitutions immediately emerge as a potential explanation for the observed variability. This correlation will be later explored in detail (see chapter 4.5.)

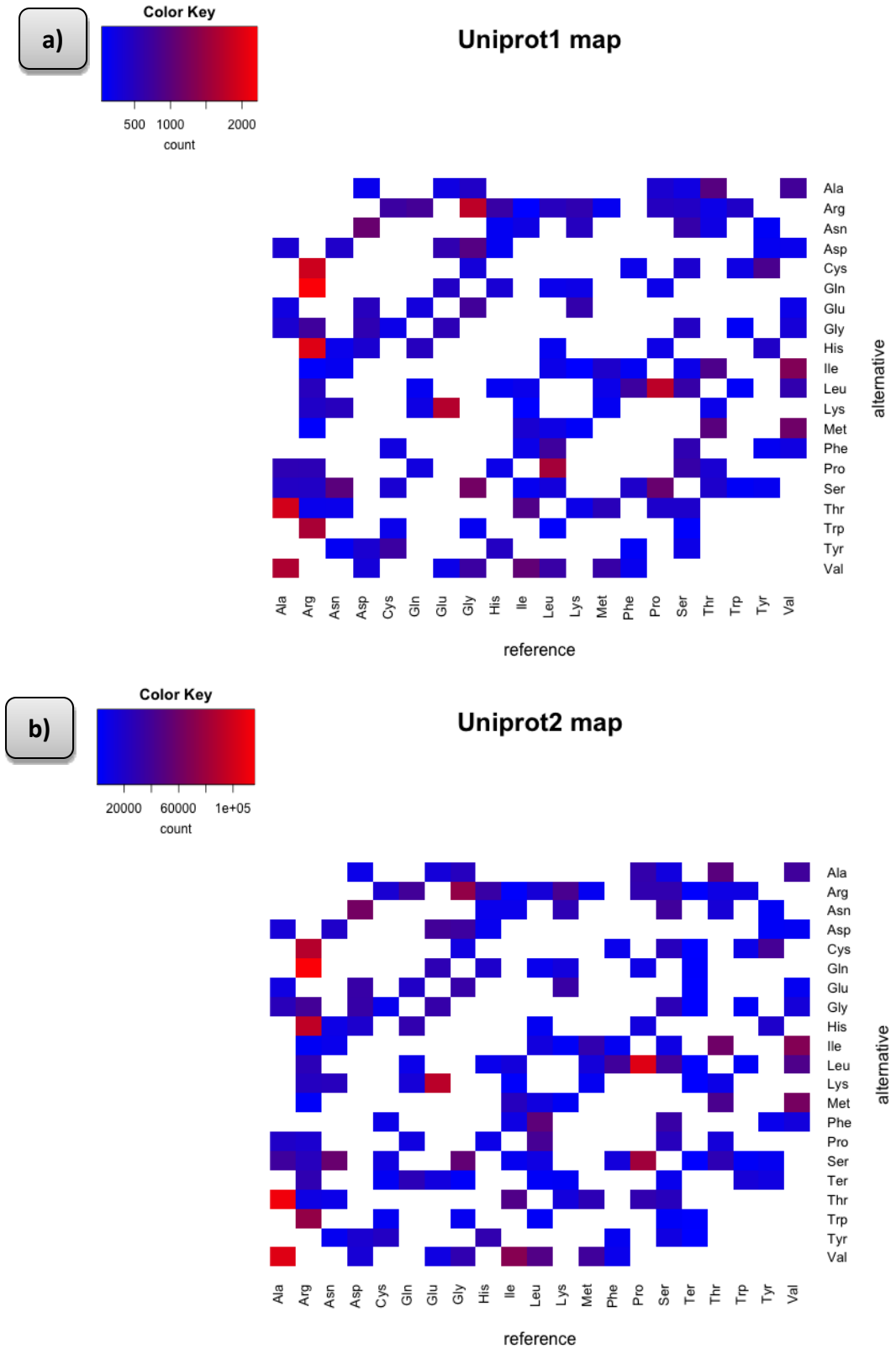


Figure 4.1: Map of amino acid substitutions. **a)** Uniprot1, **b)** Uniprot2. The general fashion and the color scheme of all subsequent heatmaps follow the ones given here. Blue locations correspond to small and red locations to large values. Columns specify amino acids that occur in a reference sequence and rows alternative amino acids.

4.2. Disease causing variants

Both Uniprot1 and Uniprot2 datasets contain information on pathogenicity of their variants. In Uniprot1 this information is from manually curated references. In Uniprot2 it's sourced from other databases as a keyword classifier. Maps of disease causing variants were constructed by taking the percentage of entries that are associated with a disease for each amino acid substitution in individual datasets (figure 4.2; normalized heatmaps and underlying tables in extended data 2).

Several things should be noted about these maps. First, disease causing percentages differ greatly between maps and are not indicative for either one. In Uniprot1 they are too high because this data represents only a subset of known variants and is heavily biased towards disease causing ones that are of interest to researchers and therefore curated more often. Uniprot2 contains unbiased data of all known variants but lacks in their annotation. For a large part of this dataset there were no information about pathogenicity and it is expected for some part of this unknown variants to be pathogenic as well. This dataset therefore underestimates the overall disease causing percentages. Since the absolute values don't give any insight, the data can be normalized without any loss of information (extended data 2). Second, relative values between amino acid substitutions give some indication of how deleterious individual substitutions are. These are consistent between datasets even though the scales are significantly different. This is especially clear in normalized heatmaps. Some of the substitutions with the highest signals are: Trp ⇒ Ser, Arg ⇒ Pro and Cys ⇒ Phe. Third, the substitutions inside the green-lined area have significantly smaller disease casing percentages. This indicates that the amino acid type has large influence on the correct protein function and the substitutions that don't change the type of the residue are less likely to be pathogenic. This effect will be further explored later (see chapter 4.4). And fourth, the maps seem to lean towards symmetry, in a sense that the values of any substitution seems to be correlated with its opposing substitution, *e.g.*, since Trp ⇒ Cys has high disease causing percentage, Cys ⇒ Trp is likely to have it as well (and same with small percentages, *e.g.*, Arg ⇔ Lys). The significance of this effect was confirmed by the correlation simulation test for Uniprot2 dataset. Null hypothesis tested was that there is no positive association between corresponding elements in the Uniprot2 disease causing substitution map (p-value = 5.1×10^{-4}; Materials and methods).

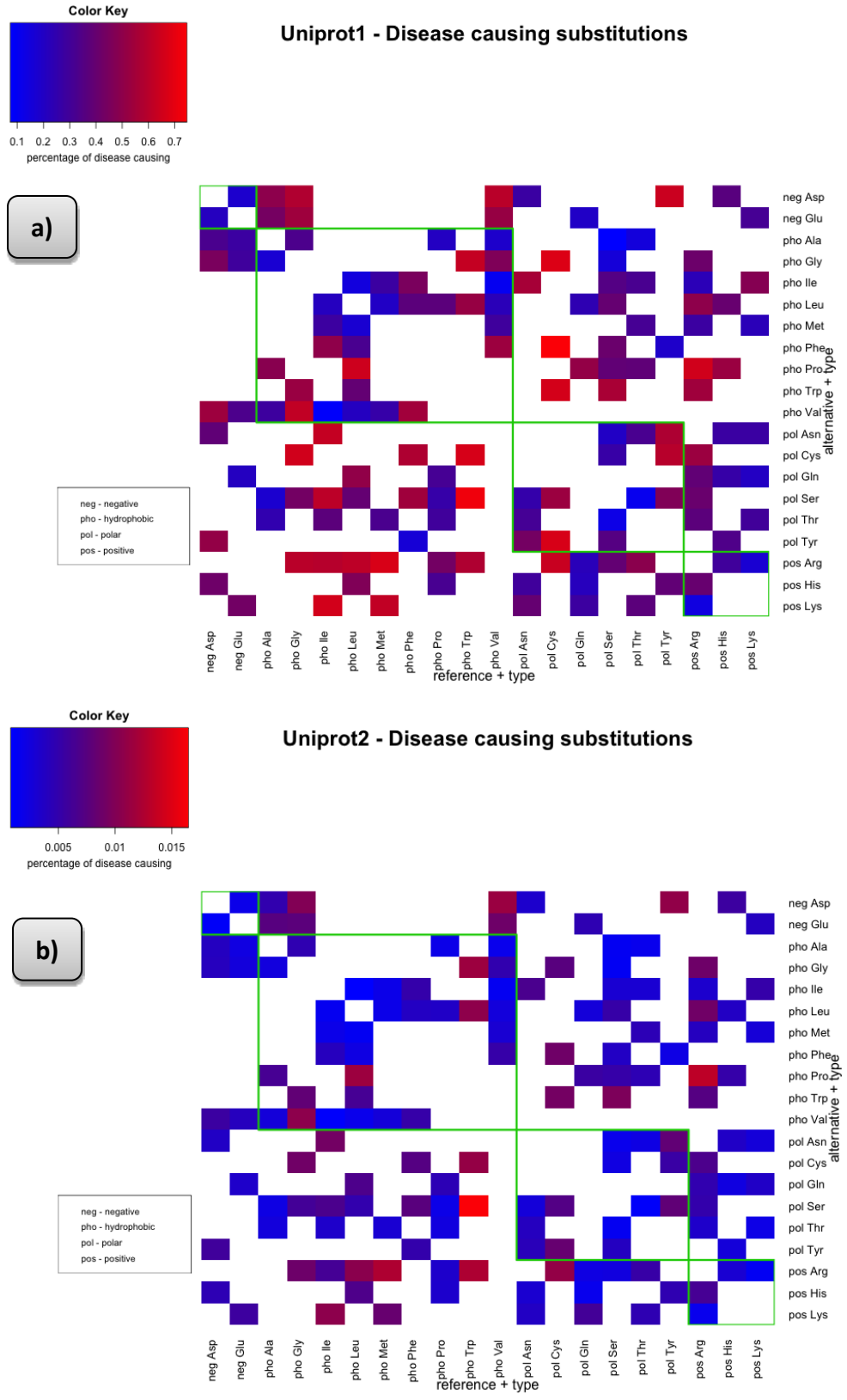


Figure 4.2: Maps of disease causing variants. **a)** Uniprot1, **b)** Uniprot2. Each amino acid is named with its 3-letter code and the type abbreviation, and grouped in such a way that the residues of the same type occur next to each other. Green lined area of the plot marks substitutions that do not change type of the residue in the primary protein structure.

4.3. Pathogenicity and occurrence frequencies of individual amino acids

On the complete map of amino acid substitutions, several significant disease associations were observed (*e.g.* Trp $\Rightarrow$ Ser, Arg $\Rightarrow$ Pro). It's interesting to look at disease causing percentages on the amino acid level of resolution (in contrast to the level of individual substitutions). To this end, pathogenicity frequencies were plotted for each amino acid in Uniprot1 and Uniprot2. Two separate bar-plots were constructed for each dataset. First is for the reference amino acids (figure 4.3) and second for the amino acids introduced by variants (figure 4.4).

For the same reasons as in the previous plots, absolute values of amino acid frequencies are not informative and only the relative values between individual amino acids are important. For the reference amino acids, tryptophan and cysteine are most often disease associated in Uniprot1 dataset while in Uniprot2, tryptophan has the highest percentage with a considerable margin. For the alternative amino acids, cysteine, tryptophan and proline have most disease causing variants in Uniprot1 while in Uniprot2 stop codon has the highest percentage. Stop codon variants are only present in the Uniprot2 dataset, as was discussed earlier. It's interesting that stop codon has a very high percentage of disease causing variants when it's introduced in genetic coding, thus causing protein truncation (4 times higher percentage than the next amino acid). When protein extension occurs, due to the removal of a stop codon, pathogenicity frequency declines (2 times lower percentage than tryptophan). This result is a reliable indication that introduction of a stop codon in genetic code has a more harmful effect than its removal. Apart from the stop codon, results are consistent with Uniprot1, showing the highest disease causing percentages when proline, tryptophan or cysteine are the alternative amino acids introduced by variants.

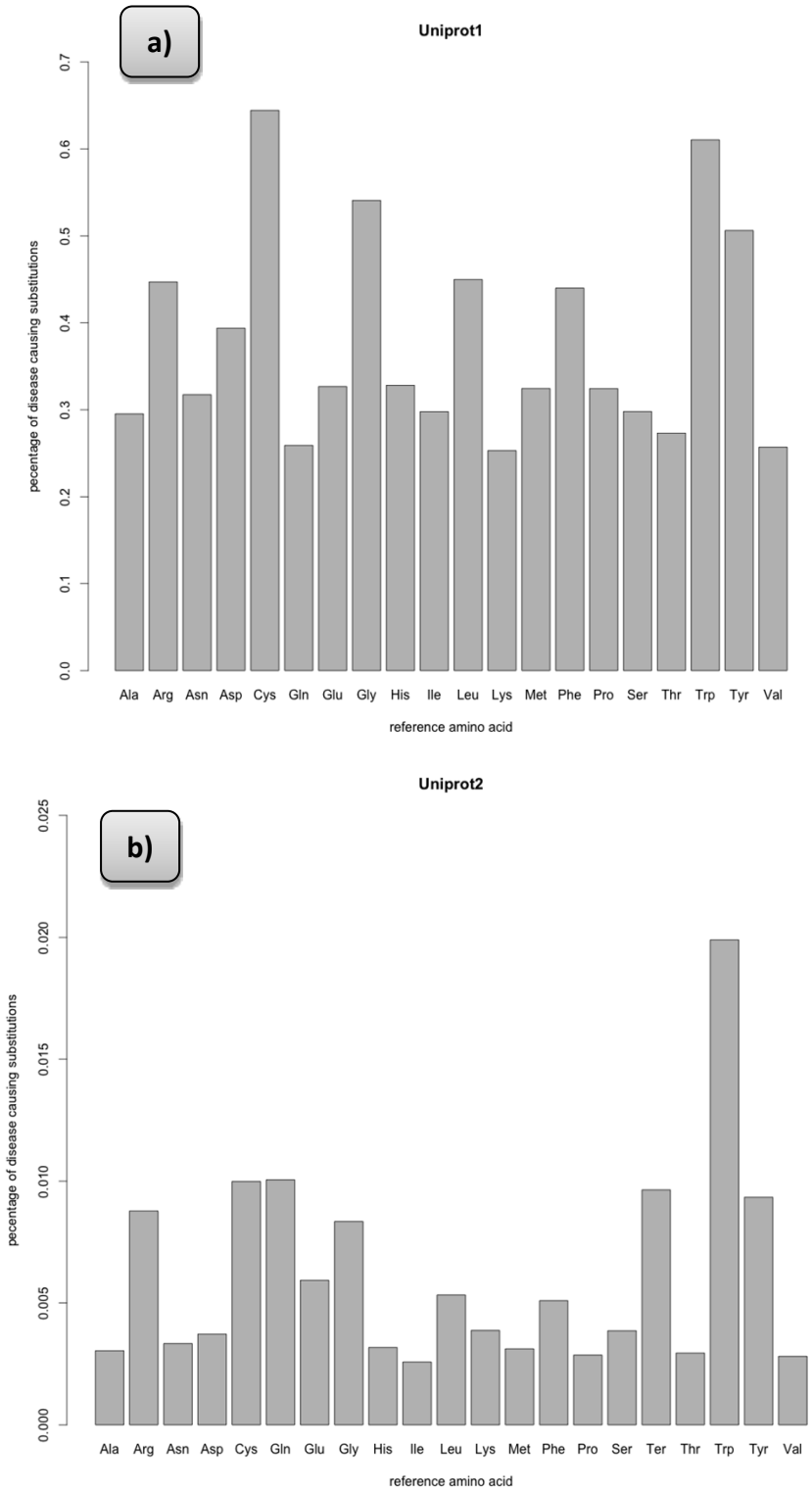
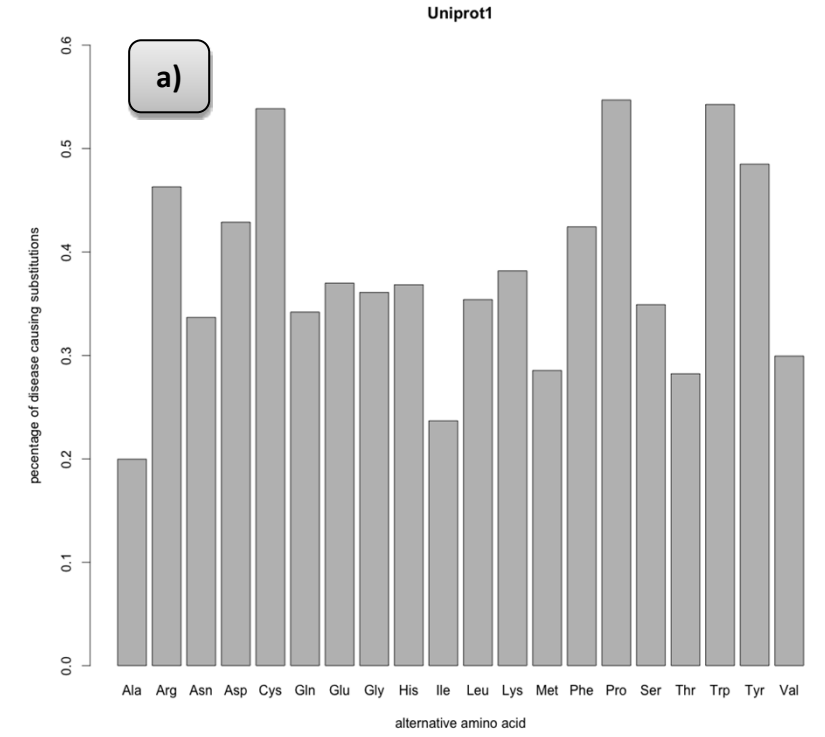
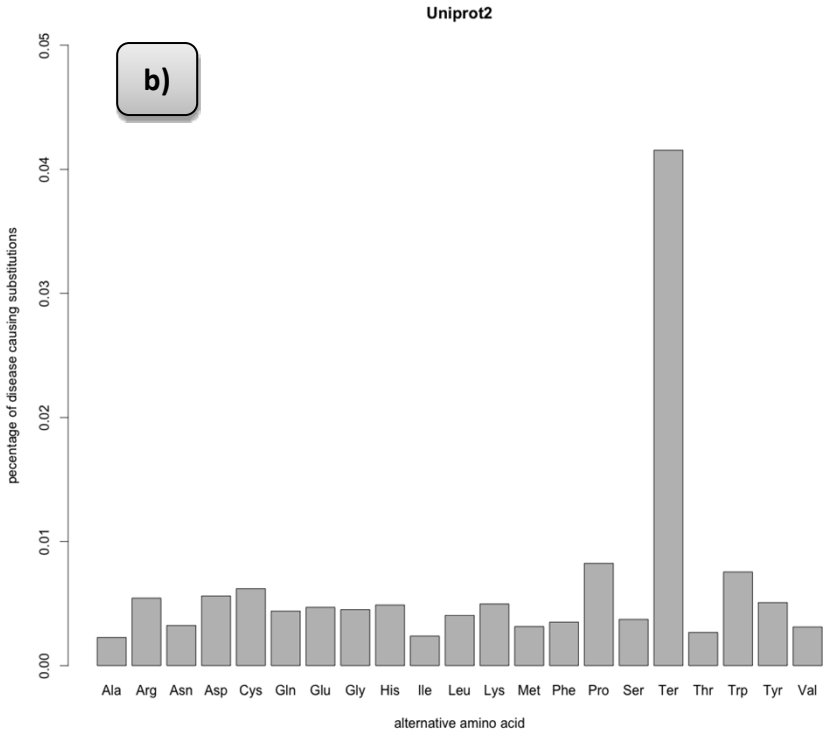


Figure 4.3: Disease causing percentages for reference amino acids, **a)** Uniprot1, **b)** Uniprot2. Each bar shows percentage of variants that are disease causing, regardless of the alternative amino acids.


Uniprot1
a)
pecentage of disease causing substitutions
0.0
0.1
0.2
0.3
0.4
0.5
0.6
Ala Arg Asn Asp Cys Gln Glu Gly His Ile Leu Lys Met Phe Pro Ser Thr Trp Tyr Val
alternative amino acid


Uniprot2
b)
pecentage of disease causing substitutions
0.00
0.01
0.02
0.03
0.04
0.05
Ala Arg Asn Asp Cys Gln Glu Gly His Ile Leu Lys Met Phe Pro Ser Ter Thr Trp Tyr Val
alternative amino acid

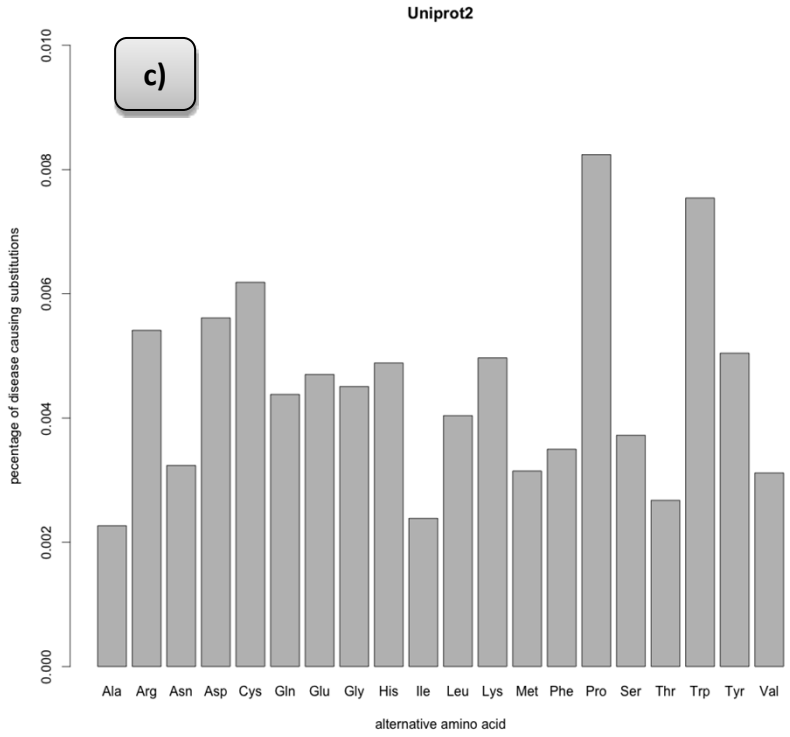


Figure 4.4: Disease causing percentages for alternative amino acids, **a)** Uniprot1, **b)** Uniprot2, **c)** Uniprot2 with stop codon variants removed. Each bar shows percentage of variants that are disease causing, regardless of the reference amino acids.

Another interesting question is how often do variants of different amino acids occur in the affected proteins, *i.e.*, what percentage of different amino acids are affected by variants (regardless of their pathogenicity). This can be answered using the information from the Uniprot1 dataset about the proteins affected by all the variants (Materials and methods). Bar-plot with the percentage of substitutions for each reference amino acid was constructed (figure 4.5). As usual, absolute values are not informative since Uniprot1 lists only part of the known variants. Relative values between amino acids show that arginine substitutions occur most often, while the amino acid with least recorded variants in human proteome is lysine.

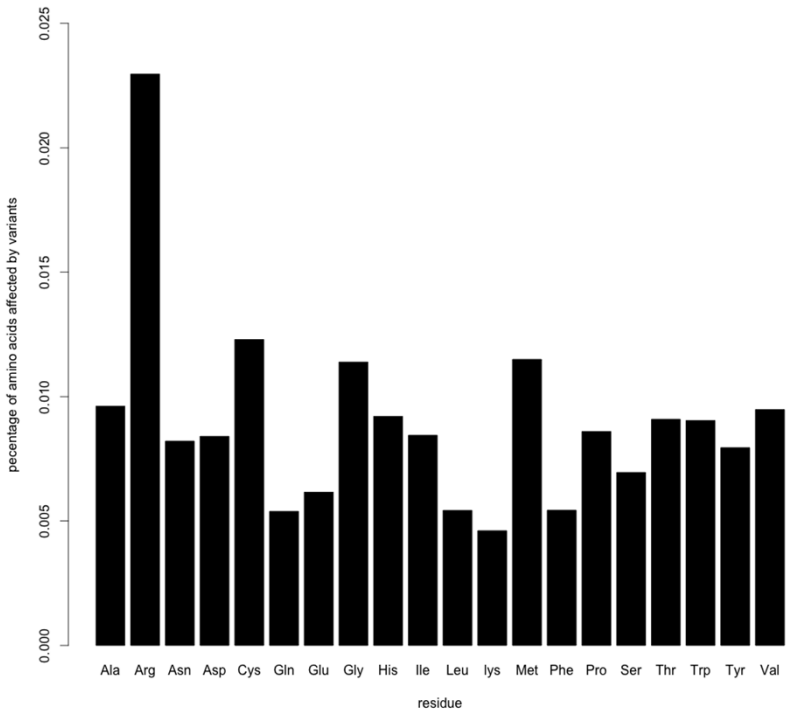


Figure 4.5: Frequency of amino acid substitutions in human proteome.

4.4. Maps based on structural classification

The structural effects of an amino acid substitution are crucial for protein function. There are several ways in which 20 proteinogenic amino acids can be divided based on their structural elements. For this analysis, relatively simple classification was chosen in which each residue belongs to one of four groups: positive, negative, polar or hydrophobic (see chapter 2.1.2.) To analyze structural effects of the variants observed in our datasets, all amino acids were grouped based on their classification and the previous analysis was repeated with this level of resolution. Two maps were constructed for each dataset. First is the number of occurrences for each transition between groups and second is the disease percentage for each of these transitions. Results from the Uniprot2 dataset are shown (figure 4.6; Uniprot1 maps and underlying tables in extended data 3).

The number of amino acids differs between groups. Furthermore, the number of ways in which transitions between groups can occur based on the genetic code of their elements is also highly variable. Therefore, the normalization of counts is necessary. For this, the normalization matrix was used (Materials and methods). For the second map

(disease percentages) this procedure wasn't necessary since any effects that introduce discrepancy occur in both the disease causing counts and the overall counts.

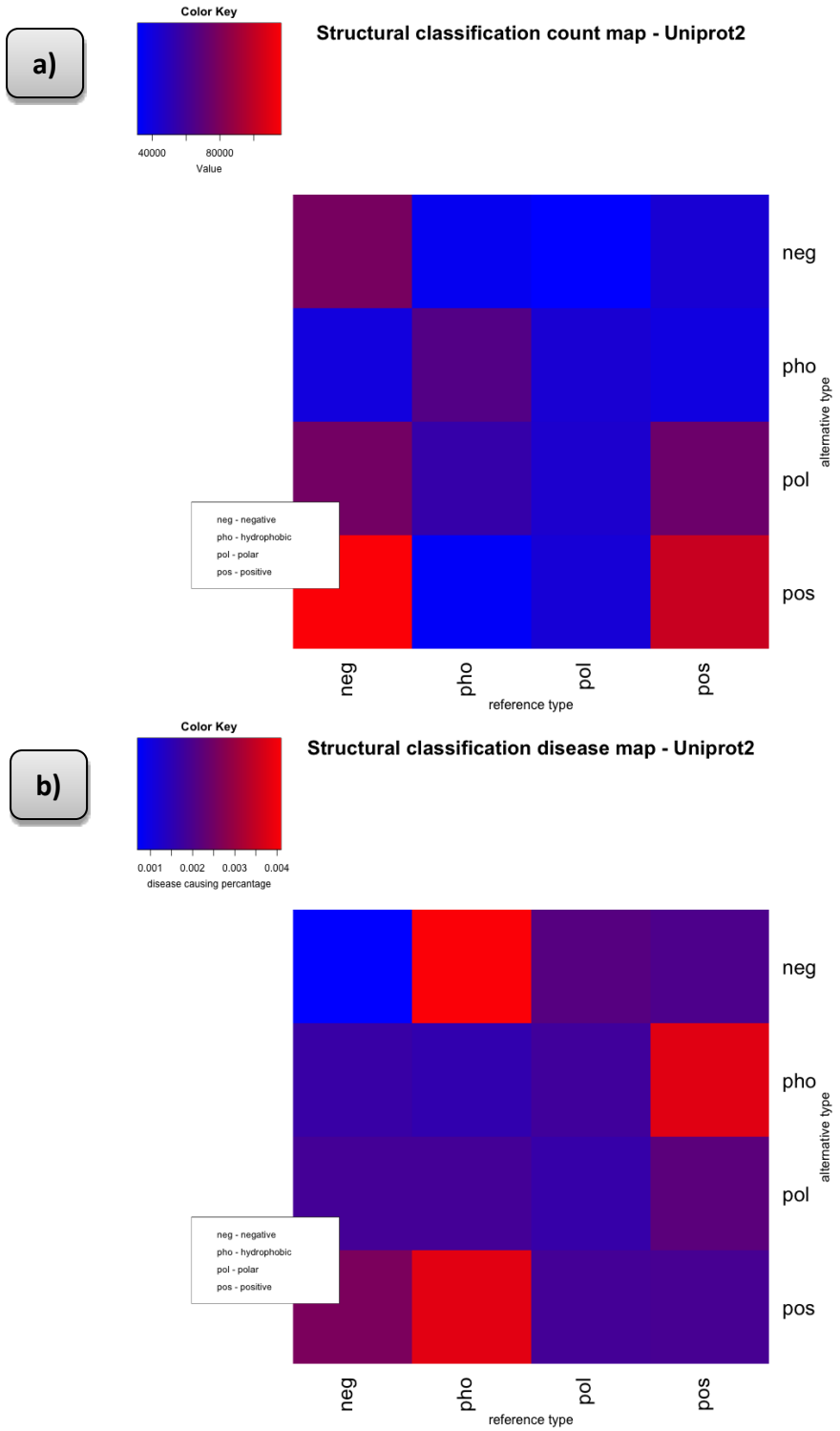


Figure 4.6: Uniprot2 maps based on structural classification, **a)** normalized counts, **b)** disease causing percentages.

Several properties of these maps should be noted. First, as with previous plots, only relative values between positions are relevant. Second, the values on the diagonal of the disease map are small indicating that the amino acid substitutions that don't change the residue type less often correspond to a pathogenic variant. This is an expected result from the biochemical perspective since the type of the residue in primary protein structure greatly influences its role in a protein. Third, three transitions that have the highest disease association are pho ⇒ pos, pos ⇒ pho and pho ⇒ neg. It is interesting to note that all of these transitions also have small relative occurrences on the count map. The same is true, although to a lesser extent and in the opposite direction, with diagonal elements - they have higher than average occurrence frequencies on the count map. These observations can be explained from the evolutionary perspective: Transitions that are often disease associated are generally under negative selection and therefore occur less frequently. The same effect is observed for the stop codon variants in the full Uniprot2 table (figure 4.1b and extended data 1) where these have exceptionally small counts, especially in the "ter" column that records variants in which stop codon is introduced and protein thus truncated. Adversely, transitions that are rarely disease causing (*e.g.* those that don't change the amino acid type) are under neutral selection, which enables their accumulation and subsequently produces more such variants in the genetic code. However, this effect isn't entirely consistent. neg ⇒ pos transition has high occurrence frequency but is also relatively often disease associated (as expected biochemically). And fourth, as with other plots, maps from Uniprot1 and Uniprot2 datasets are consistent between themselves and all discussed characteristics are evident on both.

4.5. Codons in the genetic code influence the amino acid distributions

Amino acid sequence of every protein is defined by its underlying genetic code. Three consecutive positions of this code represent one codon, which determines its corresponding amino acid. Genetic code is degenerate. Therefore, most amino acid substitutions can occur in multiple different ways. For the purpose of this analysis, only substitutions that can occur as a consequence of a SNP (that is, by one base change in their codon) will be considered. These are by far the most numerous in Uniprot1 dataset and the only ones recorded in

Uniprot2. From genetic code table, frequencies of amino acid substitutions were derived (table 2). These were counted by mapping all possible genetic code changes to their corresponding amino acid substitutions. Evidently, most diagonal elements of this table have a non-zero value. Since our datasets do not contain information on synonymous substitutions, the diagonal of this matrix was nulled wherever appropriate (*e.g.* for statistical tests) and this won't be emphasized further. Since this table contains all possible substitutions, normalizing it produces their corresponding probabilities. That is, if a random SNP occurs, causing amino acid substitution, and without any other influencing factors, probability for each variant is given by this normalized table.

Table 2: Frequencies of amino acid substitutions based on the genetic code table.

	Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	Ile	Leu	lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val
Ala	12	0	0	2	0	0	2	4	0	0	0	0	0	0	4	4	4	0	0	4
Arg	0	18	0	0	2	2	0	6	2	1	4	2	1	0	4	6	2	2	0	0
Asn	0	0	2	2	0	0	0	0	2	2	0	4	0	0	0	2	2	0	2	0
Asp	2	0	2	2	0	0	4	2	2	0	0	0	0	0	0	0	0	0	2	2
Cys	0	2	0	0	2	0	0	2	0	0	0	0	0	2	0	4	0	2	2	0
Gln	0	2	0	0	0	2	2	0	4	0	2	2	0	0	2	0	0	0	0	0
Glu	2	0	0	4	0	2	2	2	0	0	0	2	0	0	0	0	0	0	0	2
Gly	4	6	0	2	2	0	2	12	0	0	0	0	0	0	0	2	0	1	0	4
His	0	2	2	2	0	4	0	0	2	0	2	0	0	0	2	0	0	0	2	0
Ile	0	1	2	0	0	0	0	0	0	6	4	1	3	2	0	2	3	0	0	3
Leu	0	4	0	0	0	2	0	0	2	4	18	0	2	6	4	2	0	1	0	6
lys	0	2	4	0	0	2	2	0	0	1	0	2	1	0	0	0	2	0	0	0
Met	0	1	0	0	0	0	0	0	0	3	2	1	0	0	0	0	1	0	0	1
Phe	0	0	0	0	2	0	0	0	0	2	6	0	0	2	0	2	0	0	2	2
Pro	4	4	0	0	0	2	0	0	2	0	4	0	0	0	12	4	4	0	0	0
Ser	4	6	2	0	4	0	0	2	0	2	2	0	0	2	4	14	6	1	2	0
Thr	4	2	2	0	0	0	0	0	0	3	0	2	1	0	4	6	12	0	0	0
Trp	0	2	0	0	2	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0
Tyr	0	0	2	2	2	0	0	0	2	0	0	0	0	2	0	2	0	0	2	0
Val	4	0	0	2	0	0	2	4	0	3	6	0	1	2	0	0	0	0	0	12

Obvious question about the previous substitution maps (figure 4.1) is can they be explained by this frequency table. To answer this, χ^2 goodness-of-fit test was performed. Null hypothesis tested was that the probabilities for variant counts observed in each dataset equal those in the substitution matrix constructed from genetic code table. Null hypothesis was confidently reject for both datasets: p-value(Uniprot1) < 10^{-323}, p-value(Uniprot2) < 10^{-323} (for the purpose of testing the Uniprot2 dataset table, stop codon variants were discarded).

4.5.1. Codon position significance

The genetic code is fine-tuned for optimization of cellular content. This is achieved by allowing some unspecific interactions between 3rd nucleotide in the mRNA codon and its corresponding nucleotide in the tRNA molecule. That way, one tRNA can recognize multiple codons and less of these molecules are required. As a consequence, an error in translation occasionally occurs on that last codon base. Genetic code compensates for this effect through high degeneracy of a 3rd codon position. Due to this degeneracy, SNPs that target 3rd base will more often be synonymous then those that target other nucleotides. This effect can be quantified by a more detailed look into the genetic code table.

By restricting possible nucleotide substitutions to each of the three codon positions, reduced versions of the variant substitution tables can be derived (extended data 4). The expected percentage of synonymous substitutions for each position can now be calculated precisely (simply dividing the diagonal of each matrix by its sum):

- P(1st base synonymous) = 5 %
- P(2nd base synonymous) = 0 %
- P(3rd base synonymous) = 72 %

All of this shows that nucleotide positions play important role in the amino acid variation. This influence will now be explored in greater detail and with respect to disease and non-disease causing variants. It might be that optimization of genetic code extends beyond individual amino acids. For example, since most 3rd nucleotide substitutions (72 %) don't cause the amino acid change, the code might be further optimized so that the remaining substitutions that do cause the change (28 %) introduce similar amino acid (*e.g.* same type). This, or any similar optimization would be seen in discrepancy of disease causing variants with respect to the position of SNPs in their underlying code. Variants in Uniprot1 and Uniprot2 datasets were analyzed based on these positions (table 3). Notably, some variants can be caused by SNPs on multiple positions (*e.g.* Arg ⇔ Ser). These substitutions were discarded for the purpose of this analysis since they couldn't have been unambiguously assigned to either group. For each group, the overall number of variants was counted and compared with the number expected from the genetic code table (Materials and methods). The difference between these was assessed by χ^2 goodness-of-fit test. Null hypothesis tested was that the probabilities for the observed variant counts equal those

expected from the genetic code. Each group was further divided into disease associated and non-disease associated substitutions (for Uniprot1 these two groups don't add up as some substitutions have unknown disease significance). Finally, χ^2 goodness-of-fit test was carried out for each pair, testing the difference between the number of observed disease variants and the number of those expected by the overall percentage of disease associated variants in each dataset (in other words - is any position especially enriched for disease associated variants). Null hypothesis thus tested was that the probabilities of disease and non-disease substitutions for a specific codon position equal those expected from the overall distribution of these substitutions in the dataset.

Table 3: Codon position significance analysis for the nucleotide substitutions in amino acid variants, **a)** Uniprot1, **b)** Uniprot2.

a)	**observed, count (expected, count)**	**disease, observed (disease, expected)**	**non-disease, observed (non-disease, expected)**	**p-value**
1st base	1652366 (1513741)	6889 (7025)	1645477 (1645341)	**0.105**
2nd base	1713909 (1741838)	8184 (7286)	1705725 (1706623)	**5.7e-26**
3rd base	241819 (352515)	608 (1028)	241211 (240791)	**2.3e-39**
p-value	**<1e-323**			

b)	**observed, count (expected, count)**	**disease, observed (disease, expected)**	**non-disease, observed (non-disease, expected)**	**p-value**
1st base	30785 (29014)	11030 (11625)	16748 (16153)	**4.6e-13**
2nd base	34346 (33386)	14095 (12940)	16825 (17980)	**1.9e-40**
3rd base	4026 (6757)	1205 (1492)	2360 (2073)	**2.0e-22**
p-value	**7.8e-270**			

Two properties of the resulting tables should be noted. First, null hypothesis that the observed variant counts equal those in the genetic code can be confidently rejected for both datasets. The interesting aspect of this result is that the 3rd nucleotide substitutions are the ones that occur less frequently than expected while both datasets show enrichment for variants that target the 1st nucleotide. And second, variants caused by SNP on the 3rd codon base are significantly depleted of disease associations in both datasets while the variants with the 2nd base SNP are enriched for these. Variants produced by SNP of the 1st codon nucleotide have slightly lower disease association then is expected in both datasets. For Uniprot2 this depletion isn't significant and for Uniprot1 it is. These results indicate that

the most dangerous nucleotide substitutions are the ones occurring on the 2nd codon position while the most often tolerated substitutions are those on the 3rd nucleotide.

4.5.2. Variants with multiple nucleotide substitutions

As previously discussed, most of the recorded amino acid variants are caused by SNP in their underlying genetic code. However, in Uniprot1 dataset, some variants can't emerge from a single nucleotide substitution, but require multiple positions of their codon to be changed. These amino acid substitutions have > 0 occurrences in the Uniprot1 dataset and 0 frequency in the table derived from the genetic code (table 2).

Similar level of genetic code optimization can be proposed for this type of variants: Those that occur due to a single nucleotide substitution should more often be tolerable, while does that only occur when multiple nucleotides are changed should correspond to significantly different amino acids (*e.g.* structurally) and more often be disease associated. To test this, all variants from Uniprot1 dataset that require multiple nucleotide substitutions were grouped based on their disease causing capacity (those that have unknown association were discarded) and the discrepancy of these counts was assessed by χ^2 goodness-of-fit test (table 4). Null hypothesis tested was that the probabilities for disease and non-disease association of variants caused by multiple nucleotide substitutions equal those expected from the overall distribution of disease and non-disease variants in the Uniprot1 dataset.

Table 4: Distribution of variants caused by multiple nucleotide substitutions in Uniprot1.

	observed	expected
dis	52	72
nondis	120	100

Contrary to the expectation, variants with multiple nucleotide substitutions showed less disease association than their SNP caused counterparts. Furthermore, this difference was relatively significant (p-value = 1.9×10^{-3}). In an effort to explain this, type transitions (which were already shown to influence the disease association) for these two groups of substitutions were explored. If there is no significant enrichment for disease in variants that require multiple nucleotide substitutions, *i.e.* the genetic code is not optimized on that level, than the structural effects of two variant groups could potentially explain the disease association discrepancy. To this end, percentages of type changing substitutions were

calculated for each group. This was done by simply counting the number of type changing substitutions in each group and dividing it by the total number of substitutions possible in that group. The results show that SNP variants cause type transition in 65 % of cases while variants from multiple nucleotide substitutions do the same in 75 %. This result is consistent with initial assumption about genetic code optimization (although its significance can't be assessed). Therefore, it doesn't explain the results of the χ^2 test, which indicates that SNP variants have higher disease association (even though they cause type transitions less frequently). It might be that neither of these effects has any real significance. To show that, χ^2 test with larger sample size would have to be conducted.

4.5.3. Synonymous variants

Nucleotide substitutions that don't change the amino acids in their translated proteins are called synonymous SNPs. 1000G dataset contains all variants discovered in the 1000 Genome Project including the synonymous ones. This information can be used to relate predictions about synonymous substitution frequencies from the genetic code with the observed frequencies. The expected percentage of synonymous substitutions can be calculated from the table of amino acid substitutions based on the genetic code (table 2) by dividing the sum of its diagonal elements (which correspond to the synonymous substitutions) with the complete matrix sum. This calculation yields 25.5 % of expected synonymous substitutions. On the other hand, the 1000G dataset contains 944,059 protein-coding region SNPs, 376,732 out of which (38.9 %) are synonymous. This significant enrichment of our genome with synonymous variants is expected from the evolutionary perspective since these are under neutral selection more often than non-synonyms, and are therefore allowed to proliferate.

4.6. SIFT and PolyPhen

SIFT and PolyPhen are bioinformatics tools that try to predict whether amino acid substitutions in primary protein structure have a negative impact on its function. That is, whether they are disease causing. This is done by computational means and various factors are considered during the assessment. SIFT is mainly focused on conservation of amino acid

residues in sequence alignments derived from closely related sequences, while PolyPhen's prediction is based on a number of features comprising the sequence, phylogenetics and structural information characterizing the amino acid change and the substitution site.

Both methods take as an input the amino acid sequence, mutation site and the alternative residue. Their output is a numerical score between 0 and 1, which is used to predict the risk of the amino acid substitution and, based on this prediction, classify it into one of the corresponding groups. For SIFT, small scores indicate disease association and the classification groups are deleterious (for scores ≤ 0.05) and tolerated (for scores > 0.05). PolyPhen has the opposite scale in which higher values correspond to disease associated variants. Its groups are benign, possibly damaging, probably damaging and unknown (for substitutions that can't be reliably predicted).

All non-synonymous variants from the 1000 Genome Project were analyzed with these two tools and the results are available in the 1000G dataset. To test the prediction accuracy, scores and classifications for some disease casing variants were explored. Since this dataset doesn't contain information on pathogenicity of its variants, this information was mapped from the Uniprot1 dataset. Uniprot1 has the most reliable classification (which is the main use of this dataset) since its entries are manually curated. 1177 variants were successfully mapped from one dataset onto another. It should be noted that most Uniprot1 variants don't have their corresponding entries in 1000G. There are two main reasons for this. As previously discussed, a significant part of Uniprot1 variants were not obtained from SNPs in the underlying genetic code of their amino acid sequences. Any such variants are missing in 1000G. And more importantly, 1000G dataset lists only the more frequent substitutions that were detected in several of the 2504 sequenced genomes. This level of resolution is imposed by the coverage depth of the sequencing method. For Uniprot1 on the other hand, most disease causing variants correspond to events involved in severe Mendelian diseases (which are of interest for manual curation and further research) and are therefore frequently missed by the resolution of the 1000 Genomes Project.

4.6.1. Statistical tests

To analyze the classification accuracy of SIFT and PolyPhen, predictions for known disease variants were compared with their background distributions. This was tested on two levels:

For numeric prediction scores and for class predictions (figure 4.7). Background distributions were constructed from all available non-synonymous variants in the 1000G dataset. The purpose of this setup was to assess the shift of score distributions and class predictions between all substitutions and disease variants (since only these were available for testing). We would expect score distributions of disease associated substitutions to be significantly skewed towards small values for SIFT and large values for PolyPhen. Conversely, deleterious and probably pathogenic classes should be enriched for in these 1177 mapped disease variants.

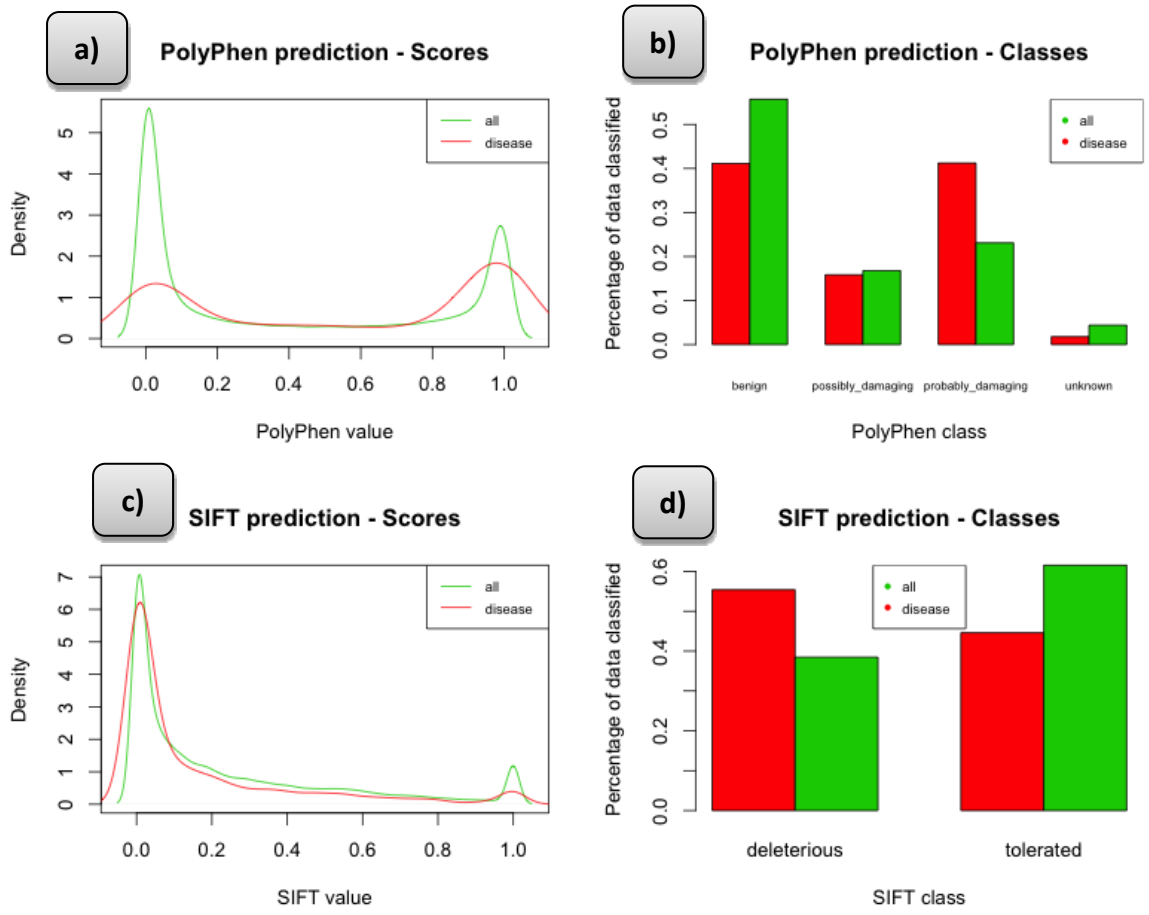


Figure 4.7: PolyPhen and SIFT scores and classifications of disease causing amino acid substitutions. **a)** PolyPhen scores, **b)** PolyPhen classes, **c)** SIFT scores, **d)** SIFT classes

It's evident, just by looking at the plots, that both methods show the expected enrichments. Their shifts were tested separately for score and class distributions. For score distributions, one-sided Wilcoxon rank sum tests were used. For SIFT scores, the null hypothesis tested was that the disease associated distribution has a location shift of its

mean from the mean of overall distribution greater than or equal to 0. For PolyPhen, the Null hypothesis was analogous with the opposite side being tested, that is, a location shift of disease sample mean from the overall mean being lesser than or equal to 0. That way, the alternative hypothesis is precisely what we expect to see based on the samples tested as it corresponds to the location shift of the mean for disease associated variants towards smaller or larger values, for SIFT and PolyPhen respectively. For both methods the null hypothesis was confidently rejected: p-value(SIFT) = 2.7×10^{-38}, p-value(PolyPhen) = 2.9×10^{-52}, thus confirming the observed shift of disease associated scores in the appropriate direction.

For class predictions, χ^2 goodness-of-fit tests were used. That way, the bias for grouping of disease associated variants into disease associated classes (deleterious and pathogenic) was assessed. The null hypothesis tested was that the classification probabilities for disease variants equal those for the complete dataset. For SIFT results this procedure is straightforward as there are only 2 groups into which variants can be classified. For PolyPhen results, only “probably damaging” and “benign” classes were used for testing, and the other two were discarded. For the “unknown” group this can easily be justified, as it contains no additional information about the prediction accuracy. The “possibly damaging” group was discarded because its precise position on the disease association spectrum is unknown and it would be hard to evaluate either the correct or the incorrect outcome of this classification. Also, having the same number of groups facilitates the comparison between methods. Null hypothesis was again confidently rejected for both methods: p-value(SIFT) = 1.3×10^{-32}, p-value(PolyPhen) = 1.3×10^{-45}, which could have been presumed from class prediction plots.

4.6.2. Comparison between methods

To assess the prediction accuracy of these two methods, their corresponding test results were compared. For Wilcoxon rank sum test, test statistics W, from which p-values were derived, are correlated with the shifts of the mean values between samples and, for same sample sizes, are comparable. This analysis shows that score distribution from PolyPhen ($W = 4.2 \times 10^{8}$, corresponding p-value = 2.9×10^{-52}) had a more significant location shift (towards larger values) for disease associated variants than score distribution from SIFT

(towards smaller values; W = 2.5×10^{8}, corresponding p-value = 2.7×10^{-38}), which corresponds to better overall classification of pathogenic substitutions.

For χ^2 goodness-of-fit test, χ^2 statistic directly corresponds to the deviation of sample probabilities from those of the underlying distribution. Therefore, larger χ^2 value, for the same number of groups (degrees of freedom), corresponds to the larger divergence of group probabilities in the tested sample. Again, the results of PolyPhen classification (χ^2 = 200.95, corresponding p-value = 1.3×10^{-45}) indicate better prediction accuracy for disease associated variants than those of SIFT classification (χ^2 = 141.39, corresponding p-value = 1.3×10^{-32}).

It should be emphasized that these tests only give the indication of prediction accuracies for the two methods. Some uncertainty was introduced by discarding one of the PolyPhen classes and, depending on the treatment of this group, the results might change. Also, in both assessments only the disease causing substitutions were tested. That is, the shift of the overall distribution from the distribution of disease associated variants. A more complete assessment would be the one that includes both a reliable set of disease and non-disease associated variants and then also looks for the false positive and false negative rates of these two methods.

4.7. Population analysis

1000 Genomes Project analyzed samples from 26 populations arranged into 5 population groups: African (afr), American (amr), East Asian (eas), European (eur) and South Asian (sas). 1000G dataset contains individual variant frequencies for each of these 5 groups. It should be noted that for a small part of the dataset (~5000 variants), population frequencies exceeded 50% in some or all of the populations. This is because the reference/alternative classification doesn't always correspond to more/less frequent, which can be seen from the distribution of alternative allele frequencies on the whole genome scale (figure 4.8). These substitutions were discarded from further analysis as they would, due to their dominant occurrence frequencies, almost entirely shape all distributions and cover up the majority of population variability.

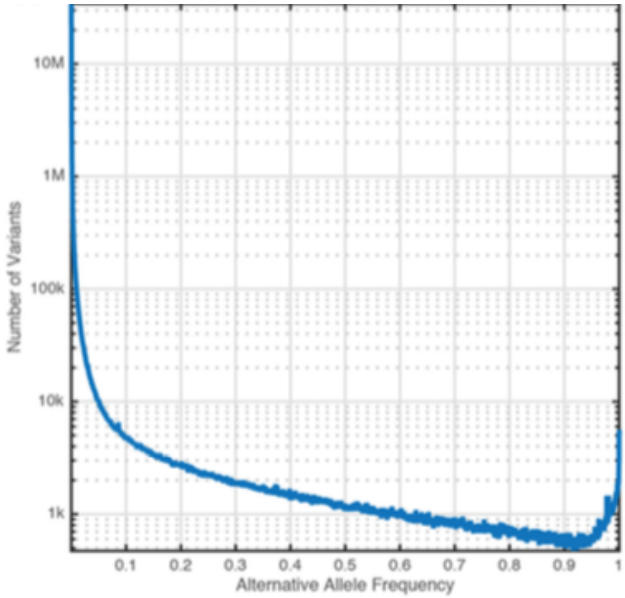


Figure 4.8: Distribution of alternative allele frequencies for structural variants in the whole genome.[31]

The results were analyzed by looking at the variability of individual amino acid substitutions between populations. To this end, the percentage of every amino acid substitution in each population group was calculated (Materials and methods). This procedure enables looking at the variation of individual substitutions between and within population groups.

4.7.1. Between population variations

Maps of frequency resolved amino acid substitutions were constructed for each of the five population groups (figure 4.9; underlying tables in extended data 5). Therefore, these plots contain no information about the absolute frequencies of individual substitutions, but rather show the relative variations between population groups.

First thing that can be observed is that the African population group differs from the rest having the exceptionally high percentages for almost all substitutions. This effect was already observed on the whole genome scale in the 1000 Genome Project, which established that African genomes contain particularly high number of variants (figure 2.13). It's reassuring that the same effect can be observed by looking only at the SNP variants in the protein-coding regions of the genome. In other population groups, no trends can be observed. However, there are several signals that stand out. Since the scale normalization

reduces the capacity for signal separation on individual maps, these will now be explored in detail on more appropriate plots.

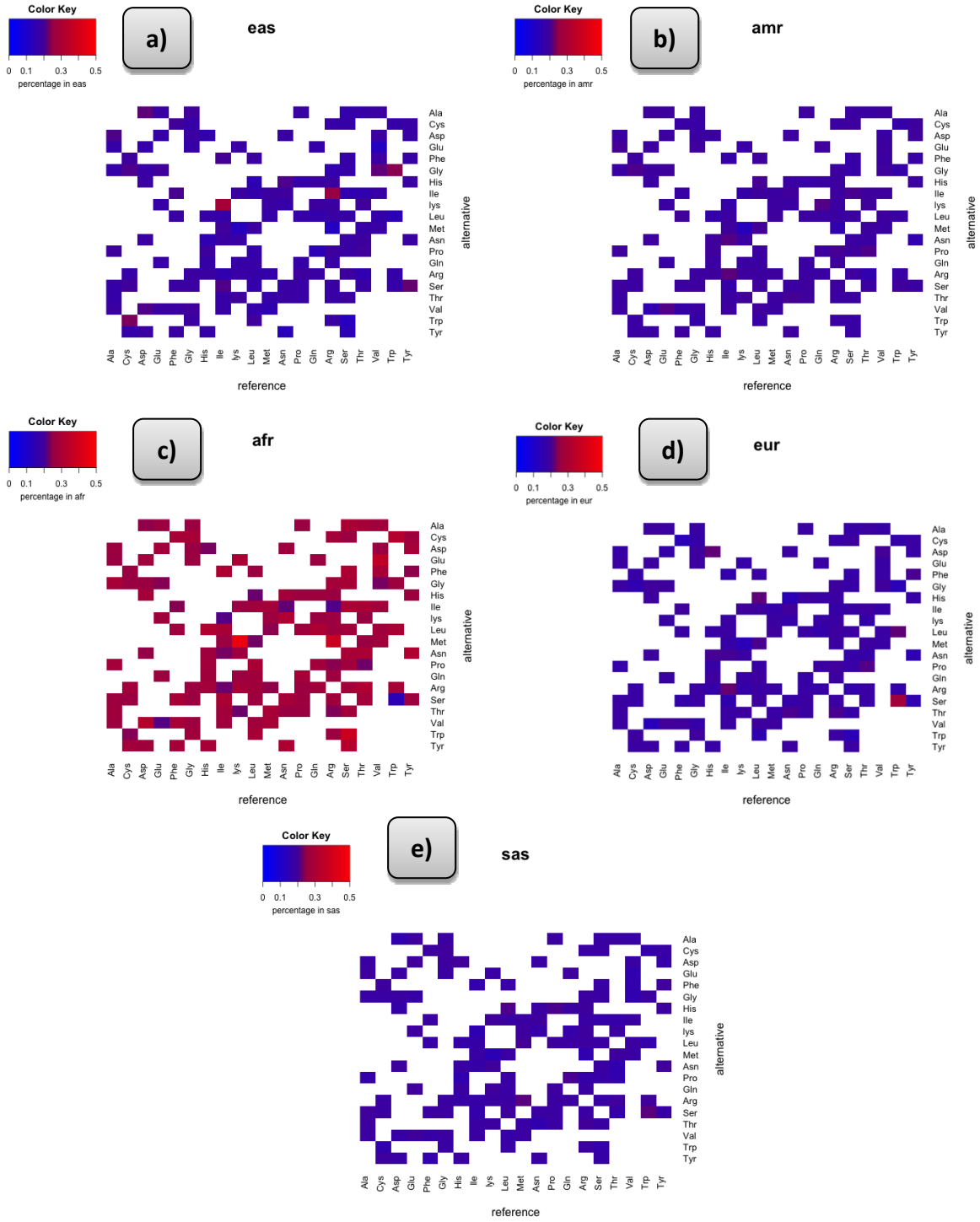


Figure 4.9: Maps of frequency resolved amino acid substitution percentages in each population group, **a)** East Asian, **b)** American, **c)** African, **d)** European, **e)** South Asian. Scale was equalized on all plots. As usual, blue indicates low substitution percentages and red high. Every position shows the portion of the corresponding substitution in that population group.

4.7.2. Within population variations

For each heatmap from the previous plot, substitution results were plotted on a separate density graph (figure 4.10). This way, the outliers can be most easily discerned. Each plot corresponds to one population group and on it, the proportions of all substitutions in that group are shown. The outliers are labeled with the amino acid substitution they correspond to.

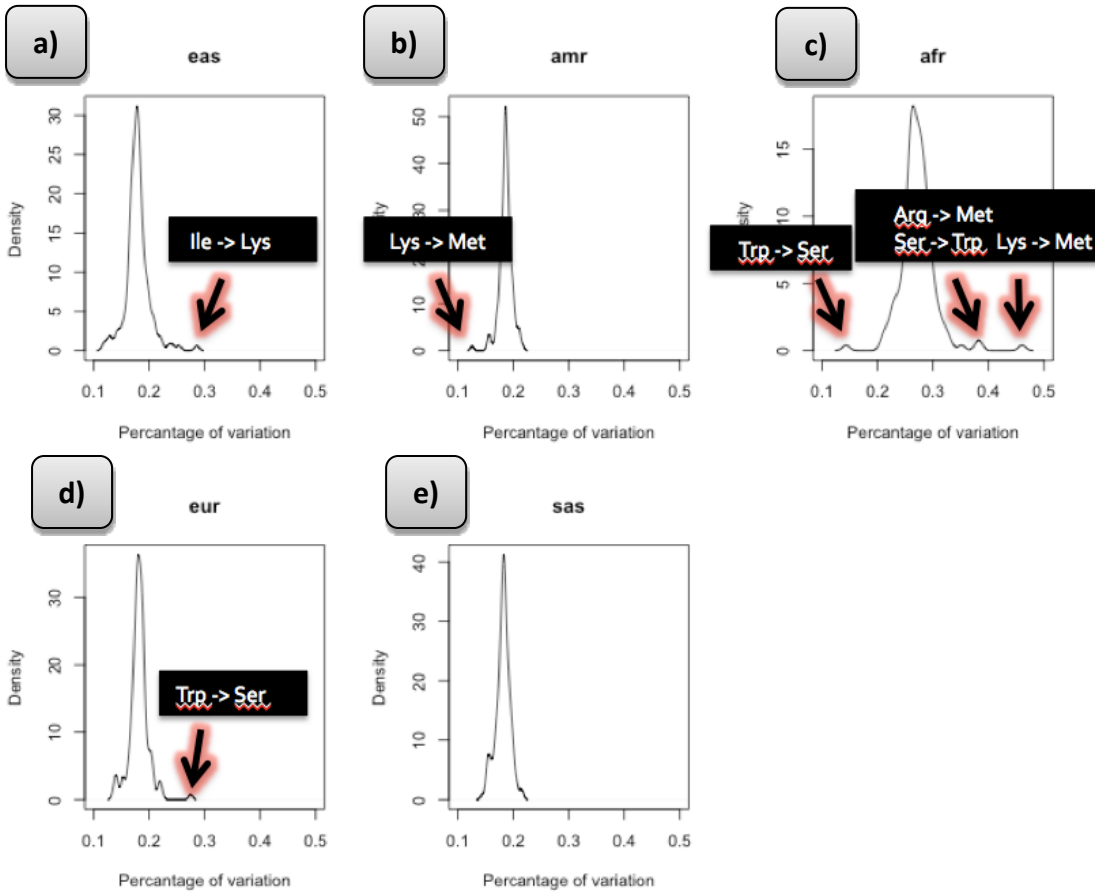


Figure 4.10: Density plots of frequency resolved amino acid substitution percentages in each population group. **a)** East Asian, **b)** American, **c)** African, **d)** European, **e)** South Asian

Interesting aspect of these plots is that for most substitutions that have either particularly low or high signal in some population group there exists the opposite trend in a different group. Lys ⇒ Met substitution has a significantly low percentage in American populations, but is enriched for in Africans. Trp ⇒ Ser has a low signal in Africans, but a high one in European population group. This characteristic is unexpected since it would be statistically more likely for some substitution that is depleted in one population group to be

equally enriched for in all the others. It might, therefore, be a consequence of some underlying biochemical discrepancy between populations. Other signals, which don't show this behavior are: Ile ⇒ Lys that is significantly more frequent in East Asians and Arg ⇒ Met, as well as Ser ⇒ Trp that are slightly enriched for in African population group.

4.8. Disease association of amino acids with combined substitution order

The relative symmetry of disease causing variants map showed a significant correlation between substitutions with the exchanged amino acid order (see chapter 4.2). Furthermore, as shown in the previous chapter, notation for the reference and alternative residue doesn't always correspond to more and less frequent amino acid. These two reasons suggest combining the corresponding substitutions in the analysis of disease associated variants to look at the amino acid pairs that have high or low disease causing frequencies instead of the individual substitutions. To this end, the ranked table of amino acid pairs was constructed, sorted by the combined disease causing percentage for both amino acid substitutions in each pair (A ⇒ B and B ⇒ A). The ordering was done by the average rank of the corresponding pair in Uniprot1 and Uniprot2 datasets, where the low ranks correspond to high disease causing percentages (table 5).

Table 5: Disease association ranking of amino acid pairs.

	amino acid 1	amino acid 2	disese causing percentage, Uniprot1	rank, Uniprot1	disease causing percentage, Uniprot2	rank, Uniprot2	average rank
1	Trp	Cys	0.66	3	0.0104	2	2.5
2	Trp	Ser	0.66	4	0.0127	1	2.5
3	Gly	Cys	0.66	2	0.0085	8	5.0
4	Phe	Cys	0.68	1	0.0082	10	5.5
5	Met	Arg	0.56	9	0.0097	3	6.0
6	Trp	Gly	0.57	8	0.0092	5	6.5
7	Val	Gly	0.57	7	0.0088	7	7.0
8	Ile	Asn	0.60	6	0.0080	11	8.5
9	Leu	Arg	0.56	13	0.0095	4	8.5
10	Gly	Arg	0.55	17	0.0088	6	11.5
11	Tyr	Cys	0.63	5	0.0065	19	12.0
12	Cys	Arg	0.56	12	0.0074	13	12.5
13	Lys	Ile	0.56	10	0.0070	16	13.0
14	Trp	Arg	0.55	15	0.0079	12	13.5
15	Val	Asp	0.56	11	0.0068	17	14.0
16	Pro	Arg	0.55	14	0.0071	15	14.5
17	Gly	Asp	0.53	19	0.0072	14	16.5
18	Trp	Leu	0.46	25	0.0084	9	17.0
19	Tyr	Asp	0.55	16	0.0068	18	17.0
20	Pro	Leu	0.50	20	0.0058	22	21.0
21	Tyr	Asn	0.50	21	0.0059	21	21.0
22	Met	Lys	0.47	23	0.0059	20	21.5

23	Val	Phe	0.54	18	0.0052	27	22.5
24	Val	Glu	0.42	32	0.0056	23	27.5
25	Gly	Glu	0.43	29	0.0052	29	29.0
26	Ser	Phe	0.46	24	0.0048	34	29.0
27	Tyr	Ser	0.40	35	0.0052	25	30.0
28	Leu	His	0.43	30	0.0050	31	30.5
29	Pro	Gln	0.43	31	0.0051	30	30.5
30	Asp	Ala	0.44	27	0.0046	35	31.0
31	Phe	Ile	0.48	22	0.0044	40	31.0
32	Lys	Glu	0.40	37	0.0052	26	31.5
33	His	Arg	0.37	42	0.0054	24	33.0
34	Ser	Leu	0.40	38	0.0052	28	33.0
35	His	Asp	0.41	34	0.0049	33	33.5
36	Ser	Ile	0.46	26	0.0043	41	33.5
37	Thr	Arg	0.44	28	0.0044	39	33.5
38	Leu	Gln	0.39	39	0.0046	37	38.0
39	Pro	His	0.42	33	0.0040	44	38.5
40	Glu	Ala	0.36	43	0.0046	36	39.5
41	Ser	Gly	0.36	46	0.0045	38	42.0
42	Ser	Arg	0.40	36	0.0036	49	42.5
43	Ser	Cys	0.39	40	0.0038	47	43.5
44	Ile	Arg	0.36	45	0.0041	43	44.0
45	Pro	Ala	0.38	41	0.0037	48	44.5
46	Gln	Arg	0.34	50	0.0042	42	46.0
47	Lys	Gln	0.24	62	0.0049	32	47.0
48	Tyr	His	0.36	44	0.0034	52	48.0
49	Asp	Asn	0.35	47	0.0035	50	48.5
50	Thr	Met	0.32	54	0.0039	46	50.0
51	Thr	Ile	0.35	48	0.0032	55	51.5
52	Lys	Asn	0.34	51	0.0032	56	53.5
53	Phe	Leu	0.35	49	0.0029	60	54.5
54	His	Asn	0.29	57	0.0034	53	55.0
55	Thr	Lys	0.33	53	0.0031	57	55.0
56	Thr	Pro	0.34	52	0.0031	58	55.0
57	Gly	Ala	0.26	60	0.0035	51	55.5
58	Glu	Gln	0.20	67	0.0039	45	56.0
59	Thr	Asn	0.31	55	0.0028	61	58.0
60	Val	Met	0.28	58	0.0029	59	58.5
61	Ser	Pro	0.31	56	0.0028	62	59.0
62	Val	Ala	0.25	61	0.0024	63	62.0
63	Tyr	Phe	0.18	71	0.0033	54	62.5
64	Met	Ile	0.28	59	0.0019	69	64.0
65	Ser	Asn	0.23	64	0.0023	65	64.5
66	Thr	Ala	0.22	66	0.0023	64	65.0
67	His	Gln	0.24	63	0.0019	68	65.5
68	Val	Leu	0.22	65	0.0022	66	65.5
69	Glu	Asp	0.19	68	0.0015	71	69.5
70	Met	Leu	0.18	69	0.0017	70	69.5
71	Ser	Ala	0.14	73	0.0020	67	70.0
72	Lys	Arg	0.16	72	0.0015	72	72.0
73	Leu	Ile	0.18	70	0.0012	75	72.5
74	Thr	Ser	0.12	74	0.0012	73	73.5
75	Val	Ile	0.10	75	0.0012	74	74.5

Second and third column show the amino acid pair. Fourth column is the combined disease causing percentage of variants for the two possible substitutions in the Uniprot1 dataset. This was calculated by summing the number of variants that are disease causing for either substitution and dividing that by the combined count of both variants. Fifth column is the rank of these percentages in decreasing order. Sixth and seventh column are the disease causing percentage and the rank for the Uniprot2 dataset (calculated in the same way). Last column is the average of the two ranks.

The table shows a good consistency between substitution ranks in two datasets. This is especially true for the outmost ranks, that is, substitutions that are least often disease associated. However, with this level of precision, a few discrepancies can also be observed. For example, {Phe,Cys} is ranked 1st in Uniprot1 dataset but only 10th in Uniprot2. Largest difference between ranks is 30, for {Lys,Gln} variants. Mean difference between ranks is 6.6 (median 6). This result confirms the observations from previous maps. Amino acid pairs that are most often disease associated are {Trp,Ser} and {Trp,Cys}, while those that are disease associated least often are {Val,Ile}, {Thr,Ser}, {Leu,Ile} and {Lys,Arg}. All of the later correspond to substitutions that don't change the amino acid type.

4.9. Structural analysis of Trp ⇒ Ser substitutions

MD simulations have become a standard tool in analyzing conformational changes to the tertiary structure of a protein caused by amino acid mutation in its primary structure.[76] In these simulations, the potential energy surface of the protein is explored first in the un-mutated, wild type protein and afterwards in its mutated variant. To obtain an initial structure of the mutated protein variant, a substitution of interest is introduced in place of reference amino acid with the rest of the structure conserved. That way, any discrepancies in the trajectory, which tracks conformational changes of a molecule through the simulated period of time, arise from differences in the potential energy surface caused by mutation.[77] A necessary prerequisite for a protein MD simulation is the crystal structure of a wild type protein (see chapter 2.1.1). The reason for this is that MD simulations track very minute conformational changes and starting the simulations from an un-optimized conformation makes it very hard to discern conformational changes caused by the mutation of interest from all the rest.

In several steps of the preceding analysis, Trp ⇒ Ser substitution was shown to be dangerous when introduced in the human genetic code (with respect to pathogenicity frequency). The aim of this part of the research was to detect conformational changes that are mutual in proteins in which this substitution is disease causing. Any such changes might contribute to the probability of a Trp ⇒ Ser substitution being disease causing. This was done with a series of MD simulations conducted in GROMACS. All proteins which had

available crystal structure in Protein Data Bank (PDB) and a reliable pathogenicity classification for the occurring Trp ⇒ Ser substitution were simulated.

4.9.1. Simulated proteins

Proteins that were used for MD simulation were chosen from variants in the Uniprot1 dataset. This dataset has a more reliable disease and non-disease classifications, but it also has more disease causing variants, which likely translates to more proteins of interest with available PDB structure. Overall, PDB structures for 19 proteins that have a recorded Trp ⇒ Ser substitution in the Uniprot1 dataset were found (figure 4.11).

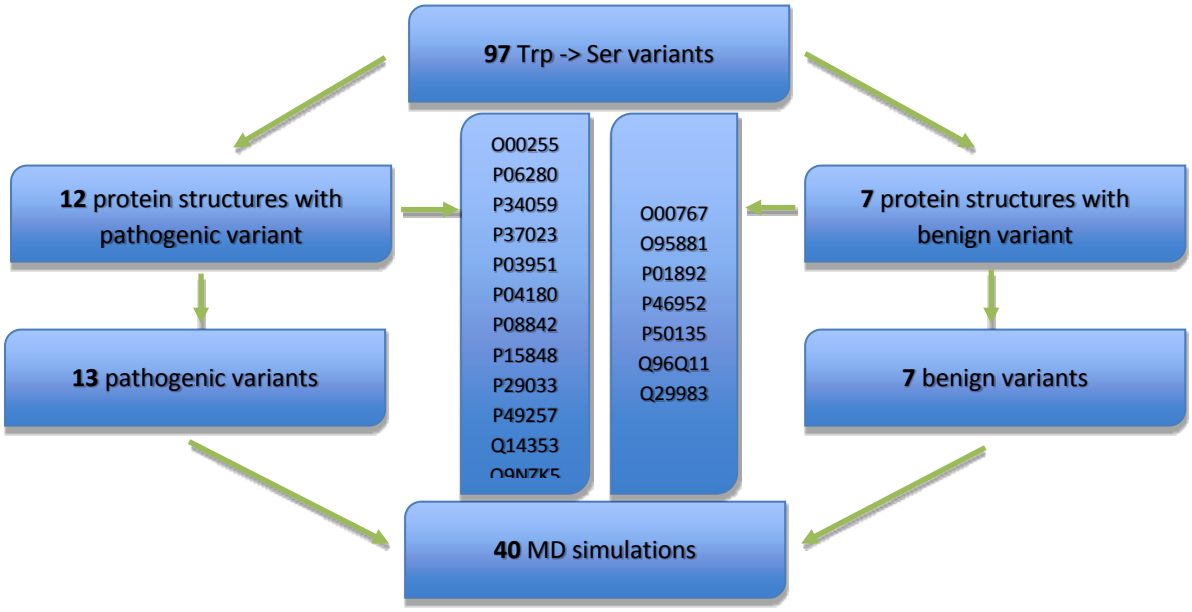


Figure 4.11: Proteins with PDB structure and a recorded Trp ⇒ Ser substitution in Uniprot1. Proteins are named with their UniProt accession numbers.

From 97 recorded Trp ⇒ Ser substitutions, 20 had the PDB structure of their corresponding protein. 13 variants were pathogenic and 7 benign. 2 pathogenic variants were from the same protein. Each of these proteins was simulated first as a wild type (Trp variant) and then with tryptophan of interest mutated into serine (table 6). In total, 40 MD simulations were performed.

Table 6: List of simulated proteins with positions of Trp ⇒ Ser substitution.

Protein accession number	Variant classification	Tryptophan position in the UniProt record of the protein
O00255	Pathogenic	188
O00255	Pathogenic	428
P06280	Pathogenic	95
P34059	Pathogenic	409
P37023	Pathogenic	399
P03951	Pathogenic	587
P04180	Pathogenic	99
P08842	Pathogenic	372
P15848	Pathogenic	146
P29033	Pathogenic	44
P49257	Pathogenic	67
Q14353	Pathogenic	20
Q9NZK5	Pathogenic	264
O00767	Benign	101
O95881	Benign	65
P01892	Benign	131
P46952	Benign	264
P50135	Benign	115
Q96Q11	Benign	239
Q29983	Benign	253

4.9.2. MD simulations

Proteins were simulated in molecular dynamics program GROMACS. The following procedure was used for all simulations:

- Proteins structures were opened and, if necessary, fixed in Swiss-PdbViewer[78] (adding any bonds or atoms there were missed by the crystal structure).
- For mutated protein variants, serine was introduced in Swiss-PdbViewer by mutating the appropriate tryptophan. Initial serine conformation was chosen in a way as to minimize the energy of the system (calculation done by the default Swiss-PdbViewer method). This step was skipped for wild type protein simulations.
- The rest of the procedure was done in GROMACS and for all relevant steps, AMBER99SB-ILDN force field was used.[79]
- Gromacs structure was generated from the final pdb file.
- Energy minimization of the whole protein was performed.

- Protein was explicitly solvated in cubic box of TIP3P water molecules (figure 4.12).
- The system was neutralized by adding the appropriate number of Na^+ or Cl^- ions.
- Energy minimization of the whole system was performed.
- NVT and NPT equilibrations were performed, each with the length of 50,000 steps and 0.002 ps time-step (adding to 100 ps of equilibration, each).
- MD simulation was performed with the 0.001 ps time-step and total length of at least 30 ns.

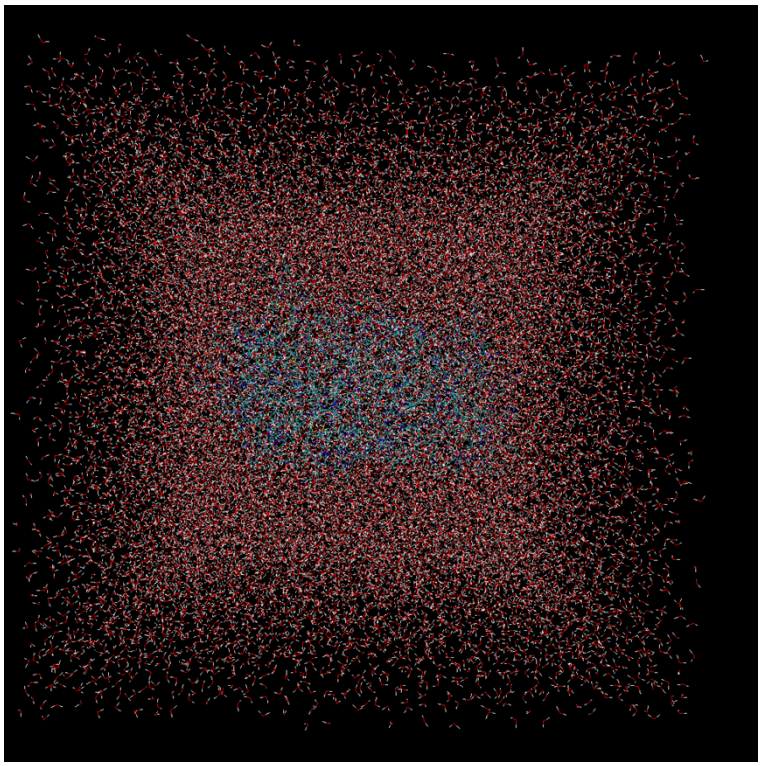


Figure 4.12: Protein P06280 solvated in a box of explicit water molecules (TIP3P).

4.9.3. Initial analysis

As previously mentioned, research on conformational changes involved in pathogenicity of Trp $\Rightarrow$ Ser variants is still in progress. Initial steps of the analysis conducted on the completed MD simulations will be described here. The objective is to find common structural elements involved in conformational changes that happen in mutated proteins of pathogenic variants, but don't happen in mutated proteins of benign variants (figure 4.13). This requires two comparisons, first between trajectories of wild and mutated proteins for each variant and then between trajectories of pathogenic and benign variants.

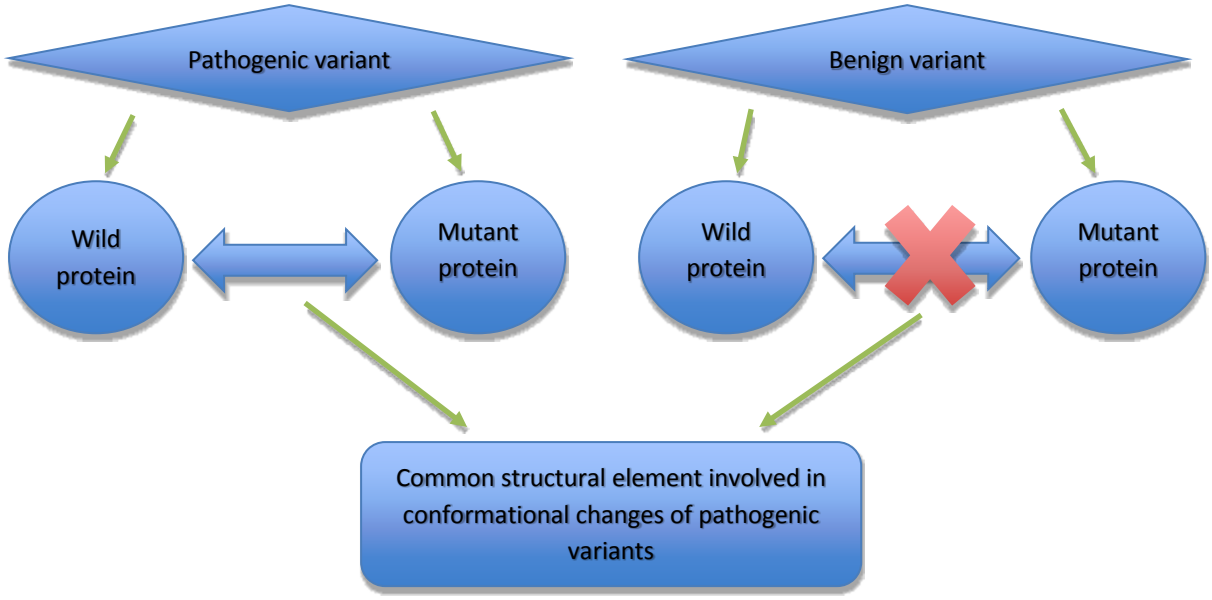


Figure 4.13: Model for Trp ⇒ Ser structural analysis.

As an initial measure of conformational changes during the course of a simulation, the relative change in distance between C-α atoms of different residues was used. This quantity was calculated with the "mdmat" function from the GROMACS program package. "mdmat" calculates the distance matrix consisting of the smallest distances between residue pairs for each simulation step and outputs an averaged matrix over the whole trajectory. For the analysis, only distances between C-α atoms were considered (since the structural differences between tryptophan and serine highly influence the measurement across the whole residue).

The output of this procedure is N x N matrix, where N is the number of residues in the initial PDB file (not necessarily the same as the number of amino acids in protein as the crystal structure doesn't usually capture the whole protein). Each position of the matrix is the average distance between C-α atoms of the corresponding residues. These matrices were calculated for all simulations, converted into numerical values and exported for the analysis.

Each individual matrix gives no information on the conformational changes occurring during the course of a single simulation (figure 4.14a), but the difference between matrices of wild and mutated variants of the same protein indicate residual contacts that differ between these two systems (figure 4.14b) and therefore likely undergo a significant conformational change in at least one of them. The focus of current analysis is on residual

contacts that include the mutated residue (figure 4.14c) and another one. This procedure can detect amino acids that show different conformational behavior in wild and mutated protein as a direct consequence of the amino acid substitution, but there are likely more conformational differences captured in other parts of the matrix that are a consequence of indirect contacts. All These matricies will be further explored and quantified in terms of distribution of amino acids that show this different conformational behaviour between Trp and Ser protein variants.

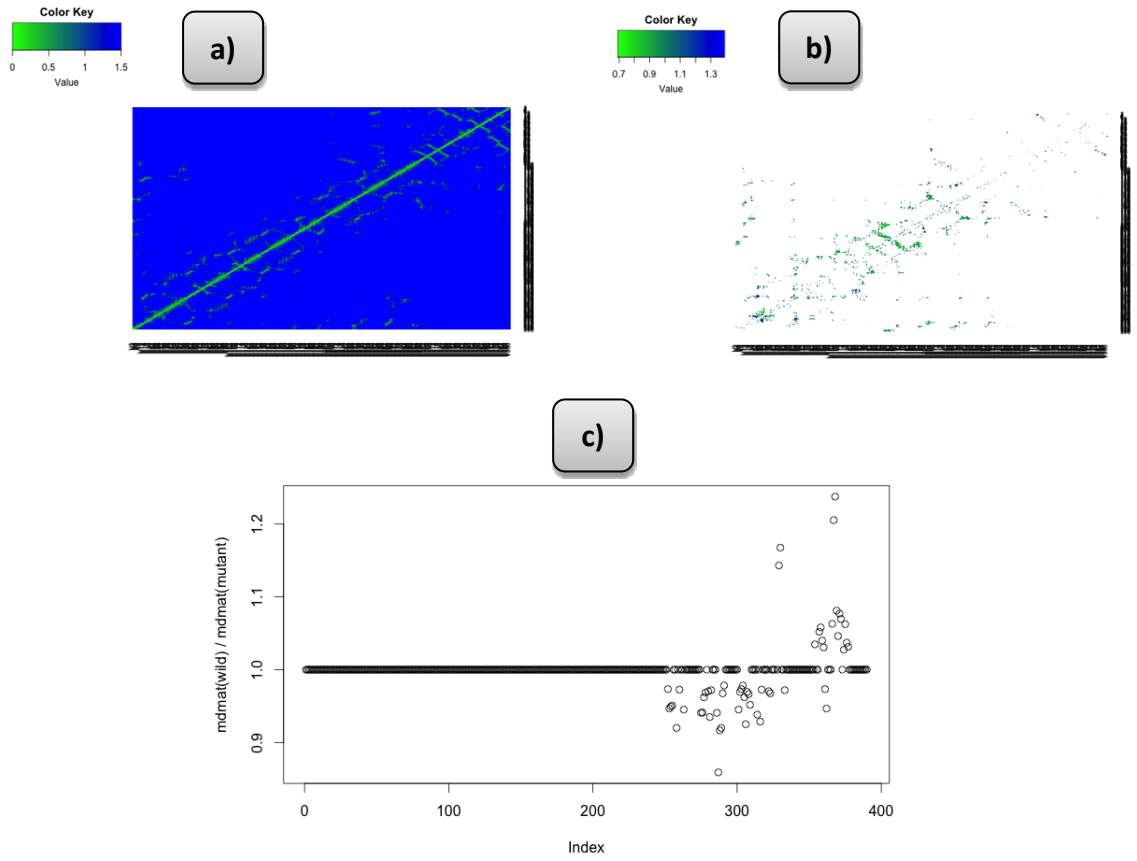


Figure 4.14: Detecting amino acids that show different conformational behavior in wild and mutated protein, **a)** "mdmat" matrix of average distances between residues in wild type protein trajectory; distances are in nm, every distance > 1.5 nm is assigned the value 1.5 (these are too far to have any influence on one another), **b)** ratio of "mdmat" matrices of wild type and mutated protein, **c)** row of the ratio matrix which corresponds to the mutated residue.

§ 5. Conclusion

Amino acid substitutions from three online available datasets were analyzed with respect to occurrence frequencies and pathogenicity frequencies. Maps were constructed for individual substitutions and for amino acid classes based on their biochemical structure. Trp ⇒ Ser, Arg ⇒ Pro and Cys ⇒ Phe amino acid substitutions were consistent between datasets as the most frequently disease associated. In terms of structural classification, pho ⇒ pos, pos ⇒ pho and pho ⇒ neg substitutions had the highest pathogenicity. The evolutionary negative selection of these substitutions was evident as they also had among lowest occurrence frequencies in all datasets.

The observed substitution frequencies were compared with those expected from the genetic code table and the null hypothesis that they can be explained by the underlying codon distribution was confidently rejected. Genetic code table was further used to test pathogenicity of variants, which showed significant depletion of pathogenic variants for substitutions targeting 3rd codon nucleotide and the enrichment of pathogenicity for variants targeting the 2nd nucleotide.

The analysis was complemented with the assessment of SIFT and PolyPhen, two bioinformatical tools used in the 1000 Genomes Project to test the pathogenicity of all nsSNP variants. Only disease causing variants that could have been confidently mapped were tested and PolyPhen showed slightly better classification accuracy of disease causing variants, both in terms of numerical score assignment and group classification.

Population specific substitution maps were constructed for 5 population groups defined in the 1000 Genomes Project. African genomes had a significant enrichment for almost all amino acid substitutions. This result was consistent with the assessment of all structural variants by the 1000 Genomes Consortium that found similar enrichment of alternative alleles in genomes from African populations as well as with the out-of-Africa model of human origin. Additionally, some unexpected discrepancies between specific substitutions were found in several population groups. African genomes were enriched with the Lys ⇒ Met substitution, while American genomes were depleted of the same. Also, European genomes were enriched with the Trp ⇒ Ser substitution and African genomes depleted of it. This might be a consequence of some underlying biochemical difference.

A significant symmetry of disease causing frequencies was detected between corresponding amino acid substitutions (A $\Rightarrow$ B & B $\Rightarrow$ A) which prompted ranking of amino acid pairs based on their average disease association. {Trp,Ser} and {Trp,Cys} were on top of these ranks while {Val,Ile}, {Thr,Ser}, {Leu,Ile} and {Lys,Arg} held the bottom. These results were consistent with the preceding analyses.

Finally, Trp $\Rightarrow$ Ser substitution was chosen for the structural analysis that was conducted through a series of MD simulations. In total, 40 protein simulations were performed on 19 different proteins with each one simulated as a wild type and with introduced mutation(s) and each simulation at least 30 ns in length. The analysis of results was initiated by the construction of distance matrices between C-α atoms of different residues. These matrices will be further used to detect structural elements involved in conformational changes specific to proteins that have pathogenic Trp $\Rightarrow$ Ser substitution.

§ 6. Materials and methods

Accessing Datasets

Data for Uniprot1 and Uniprot2 can be accessed in the archive of UniProt website.[80] File names are “humsavar.txt” and “homo_sapiens_variation.txt” respectively. Release version 2016_02 from February 2016 was used. Newer version of these datasets can be found under the current release section of the UniProt website.[81] File names are the same.

Data for the 1000G dataset can be accessed in the 1000 Genomes Project repository.[82] It contains fully annotated variant calls from the completed project (main phase 3). Variants are provided in a separate file for each chromosome.

All three datasets are also stored with the electronic version of this thesis on the accompanying DVD.

Dataset: Uniprot1

This dataset was obtained from the Universal Protein Resource (UniProt) database. UniProt is comprised of three subunits: UniProtKB contains extensively curated protein information with emphasis on function, classification and cross-references, Uniref combines closely related sequences into a single record to speed up sequence similarity searches and UniParc is a comprehensive repository of all protein sequences, consisting only of unique identifiers and sequences.

Uniprot1 contains information of manually curated human polymorphisms and disease mutations and is a part of UniProtKB subunit. It is a tab delimited text file of amino acid altering variants imported form Ensemble Variation databases.

Information contained in this dataset are: affected gene-name and UniProt entry of its corresponding protein, reference variant identifier, amino acid change in its usual format where first amino acid corresponds to the residue in the reference sequence and second, the alternative residue introduced as a consequence of the variant, type of variant (whether it’s disease causing, benign polymorphism, or unknown), SNP identifier where applicable, and finally, disease name and reference for disease causing variants. Overall, there are 73,266 entries (figure 6.1a).

Several features of this dataset should be noted. First, it contains small number of variants (significantly fewer than other datasets). Second, it is manually curated. Third, there is relatively high percentage of disease causing variants. This is expected because of the previous point; these variants are of greatest interest for further research. Fourth, significant part of the dataset is derived from direct proteomic observations and is not obtained through genome variation mapping (even though the majority of the variants are still, most likely, discovered in this way). Consequently, there are many entries that don't have their associated SNP identifiers and, more importantly, as will be seen, some of them contain amino acid substitutions that can't be obtained from SNP translations since they require multiple nucleotides in their codons to be changed. And fifth, disease causing classification for variants should be reliable (because of the second point).

This dataset was used in most of the conducted analysis. Its advantages are simple, informative layout that enables various data extraction techniques and reliable disease causing classification that contributes to the significance of any disease associated signals. Its biggest disadvantage is small sample size.

Dataset: Uniprot2

This is another UniProt dataset. It lists all, un-curated protein altering variants imported from the Ensemble database and includes: 1000 Genomes Project, Exome Aggregation Consortium, Exome Sequencing Project and Catalogue of Somatic Mutations in Cancer. Variant types listed are missense, stop lost and stop gained mutations. When available, additional phenotype or disease descriptors are imported from Ensemble and included with their corresponding variants.

In this datasets, each entry corresponds to the mapping made between UniProtKB isoform sequence and Ensembl transcript. There are two layers of redundancy in the data introduced by this procedure. First, more than one transcript can often be mapped to the same isoform sequence (due to the alternative splicing events), consequently the same variant that occurs in more than one transcript is described multiple times in the dataset. And second, due to the automated mapping pipeline, whenever a variant has more than one phenotype or disease description, each description is recorded as a separate entry and

the variant is therefore duplicated. Second redundancy is clearly undesirable and is therefore discarded. As for the first redundancy, there are arguments both for discarding and for leaving it. From genomic standpoint, it makes more sense to discard this redundancy - if some SNP variant is known to be disease associated (*e.g.* from GWAS), then it would make no sense duplicating it simply because it appeared in the protein coding region that codes for multiple isoforms. However, from proteomic standpoint there is also an argument for leaving this redundancy - maybe some isoforms affected by the variant have functionally impaired proteins and for others, the protein function is unaffected, if this discrepancy was recorded, than it should be preserved. The same analysis was conducted with this redundancy both discarded and retained and the results were nearly identical. Since the case described under the genomic standpoint is a more plausible scenario (after all, data was obtained through translation of the protein-coding genomic regions), this dataset will be used when discussing the results. Discarding both redundancies leaves the dataset with 3,843,322 variants (figure 6.1b).

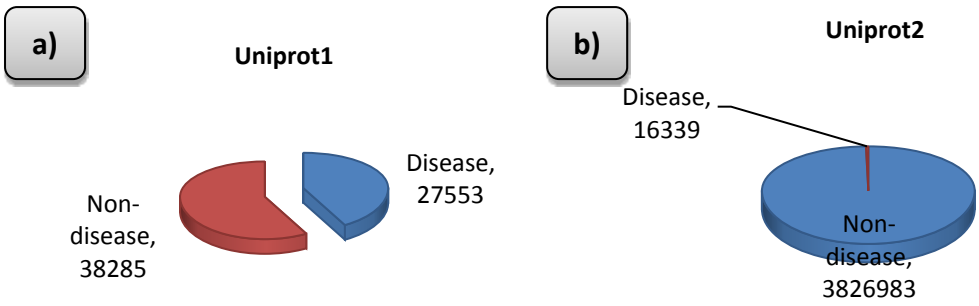


Figure 6.1: Distribution of disease and non-disease variants, **a)** Uniprot1 dataset, **b)** Uniprot2 dataset.

Corresponding features of this dataset are: First, it's ~50 times bigger than Uniprot1 and is the largest dataset available as it combines findings of 1000 Genomes with several other, similar projects. This contributes to the statistical significance of results and therefore, in almost all analysis, Uniprot2 results should be taken with the greatest confidence. Second, it was generated by the automated pipeline and its entries have not been examined. Third, it contains information on pathogenicity for some of the variants, but the classification is more ambiguous. Every variant was merged with its corresponding Ensemble entry. If there was any information available, it was represented as a keyword and

these keywords were extracted by the automated pipeline.[83] From all detected entries, the following classifiers were accepted as disease associated: "pathogenic", "not provided,pathogenic", "uncertain significance,pathogenic", "likely pathogenic,pathogenic" and "pathogenic,other". Fourth, all variants were obtained by SNP translations and, therefore, only a subset of all possible amino acid substitutions (381) can be observed - those that can occur by a single nucleotide change in their codon (150; see chapter 2.2.) In addition to this, and for the same reason, introduction and loss of a stop codon are also observed. And fifth, since this data isn't manually curated, disease classification is not as reliable as for the Uniprot1 dataset.

Advantages of the Uniprot2 dataset are its size and the inclusion of stop codon variants. Slight disadvantage is the ambiguity of disease associated variants classification. This dataset was also used in all conducted analysis.

Dataset: 1000G

Uniprot1 and Uniprot2 contain processed data, which means that the original data from the genome sequencing projects (or other sources) was retrieved and modified to enable the exploration of amino acid variants. In this process, some information was discarded. For that reason, the original 1000 Genome variant calling output contains some additional results that can be used to complement the amino acid variation analysis. The 1000G dataset contains this raw output and is used in several steps of the analysis. Notably, it contains information about variants in the whole genome and not just its protein coding regions, which is why several filtering steps were necessary to extract only the SNP causing variants. After this filtering, dataset contained 576,738 variants.

Four aspects of the 1000G dataset were explored. First, while Uniprot1 and Uniprot2 provide a comprehensive list of known amino acid variants, they do not contain their occurrence frequencies. This information is recorded in 1000G, and from it, the full map of amino acid substitutions was constructed (see chapter 4.7.) Second, 1000G also contains variant frequencies for each of the five population groups studied in the 1000 Genome Project: East Asians, South Asians, Europeans, Americans and Africans. This enabled construction of population-specific maps. Each of them also considered occurrence

frequencies for individual variants (see chapter 4.7.1.) Third, since this dataset contains sequencing information on genomic level, synonymous variants found in the protein-coding regions are also recorded. These were used to assess non-synonymous mutation frequencies (see chapter 4.5.3.) And fourth, the main disadvantage of the 1000G dataset is that it doesn't contain information on pathogenicity of individual variants. However, it includes scores of protein variant disease classification assessments from two bioinformatical tools: PolyPhen[84] and SIFT.[85] These scores were analyzed with respect to the percentage of correct classifications for known disease causing variants (see chapter 4.6.)

Disease casing map symmetry

To test for the significance of correlation between symmetrical elements in disease causing Uniprot2 map the following procedure was applied: First, Pearson correlation coefficient between the corresponding elements was calculated. Then, underlying matrix elements of the map were randomly rearranged and the correlation coefficient calculated for the new map. This procedure was repeated one million times. Finally, p-value for the significance of the initial correlation coefficient was assessed by comparing the number of coefficients from simulated matrices that had the value larger than the one from the initial matrix (one sided test, as we don't expect negative correlation). Null hypothesis tested in this manner was that, given the disease causing percentages from the Uniprot2 dataset, the association between corresponding amino acid substitutions is ≤ 0. Null hypothesis was rejected with high confidence (p-value = 5.1×10^{-4}; figure 6.2).

Notably, the p-value was slightly higher than the one calculated by the one sided Pearson correlation test which assesses the relation between corresponding elements, but without the condition imposed on their values (Pearson p-value = 2.8×10^{-4}).

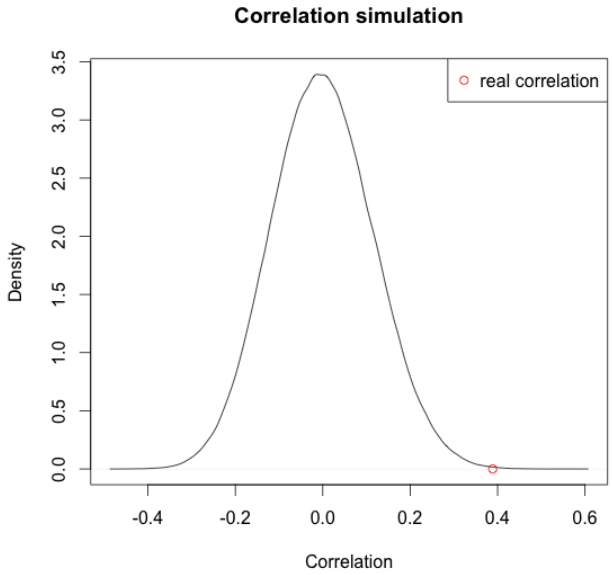


Figure 6.2: Density plot of correlation coefficients between symmetrical elements for simulated Uniprot2 disease causing variant maps.

Percentage of amino acids affected by variants

Uniprot1 dataset contains reliable references to proteins affected by all the variants. This information can be used to assess the overall fraction of substitutions that occur for different amino acids. To explore this, number of occurrences of each amino acid was counted in the primary structure of all proteins that the variants in Uniprot1 dataset affect. Fraction of each individual amino acid in the proteome that is affected by variants was calculated by counting the number of occurrences of that amino acid as the reference residue in the Uniprot1 dataset and dividing it by the number of occurrences of that residue in all protein sequences.

Amino acid sequences for all required proteins were obtained from the UniProt repository.[7]

Normalization matrix for structural classification

Since the number of amino acids in each class and the number of possible transitions between classes differ, normalization of transition counts was necessary. This was achieved

by dividing the total number of transitions with the normalization matrix (table 7). This matrix counts the number of possible transitions between groups by taking both discrepancies into account. Each value corresponds to the number of amino acid substitutions that cause the given transition and can occur as a consequence of a SNP. The second condition was imposed because the number of such substitutions greatly exceeds the ones that occur with multiple nucleotide changes in Uniprot1 dataset and are the only ones recorded in Uniprot2. Effectively, it's the number of amino acid substitutions that cause the given transition type and have occurrence frequency > 0 in the Uniprot2 dataset.

Table 7: Normalization matrix for structural classification.

	neg	pho	pol	pos
neg	2	6	3	2
pho	6	32	18	10
pol	3	18	14	10
pos	2	10	10	4

Expected number of variants for each codon position

To calculate the expected number of variants for each codon position, the sum of substitutions that correspond to variants caused by SNPs in a specific position (intersection of tables in extended data 4 and 1a) was divided by the sum of the whole substitution matrix (extended data 1a). That way, only variants that correspond to exactly one SNP position are counted. As a consequence, some substitutions are not included into any of the three sets and the percentages of counts sum to slightly less than 1.

Substitution frequencies in individual population groups

To analyze the variation of amino acid substitutions between 5 population groups, the following procedure was applied: First, all missense variants from the 1000G dataset were grouped according to the amino acid substitution they specify. Then, the average occurrence frequencies of all substitutions were calculated for each group. This step included the assumption that each population group has the same number of individuals sequenced. Since variant distributions of individual populations are almost identical, minor

deviations from this assumption wouldn't significantly affect the results. Then, the overall substitution frequencies (for all groups combined) were calculated. Finally, for every amino acid substitution, the proportion of its occurrence frequency in each population group was calculated. With this procedure, frequencies of individual variants in the 1000G dataset were taken into account, and the proportion of every amino acid substitution, in each population group, obtained. Notably, for every substitution (*e.g.* Cys $\Rightarrow$ Tyr), values in all 5 population groups sum to 1.

§ 7. Extended data

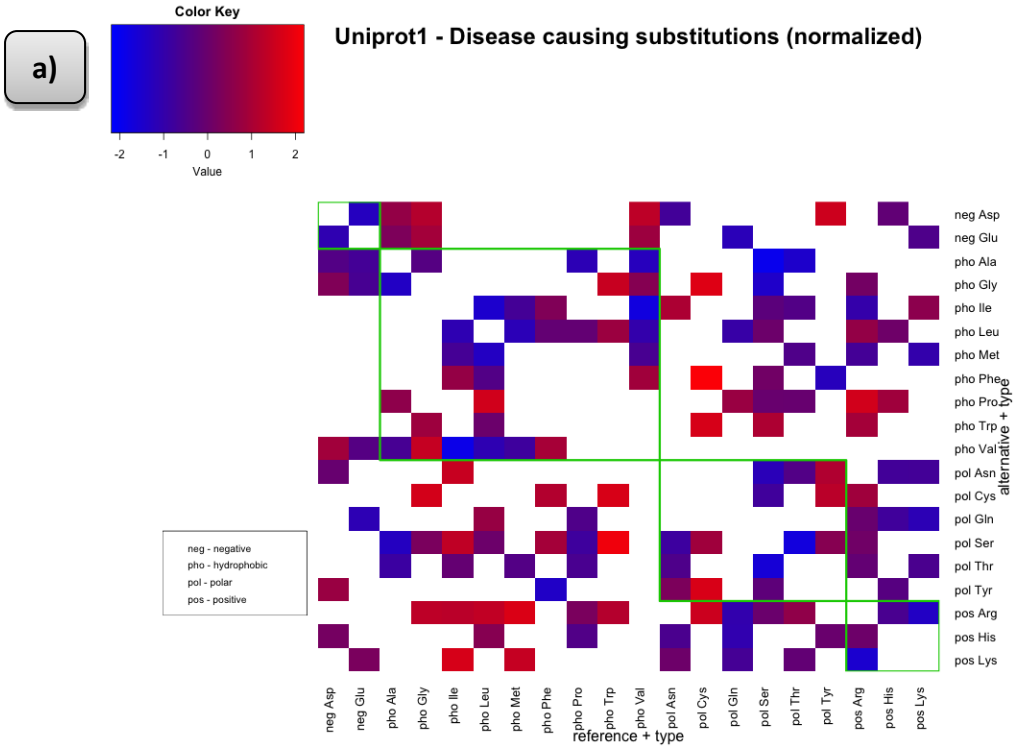
a)

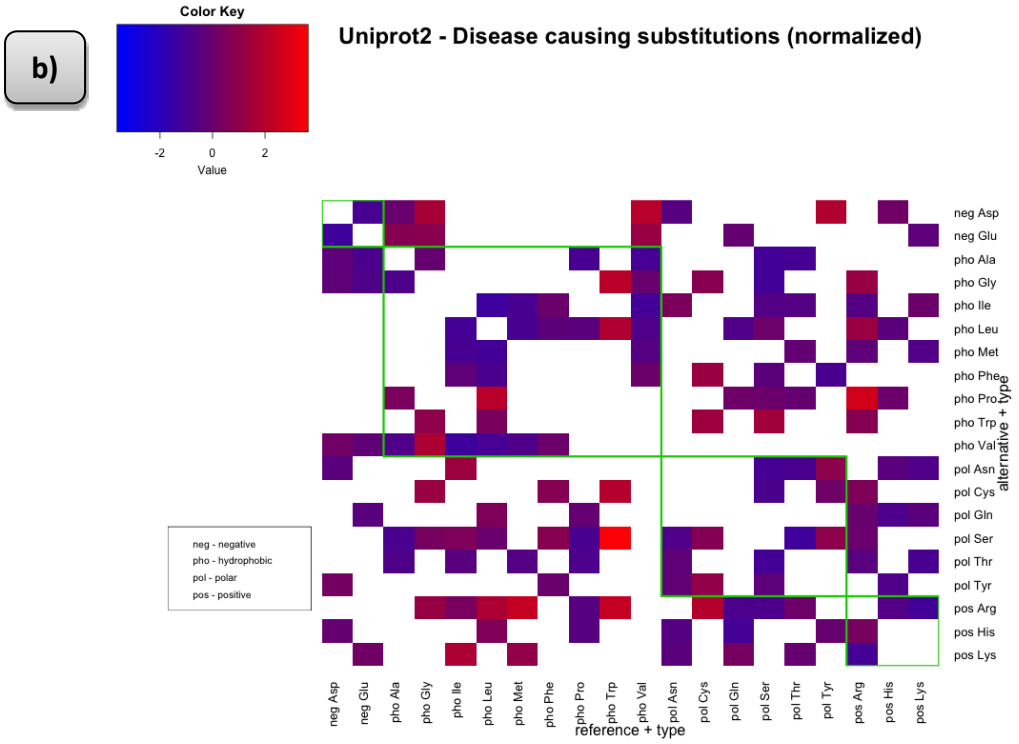
reference / alternative	Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	Ile	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val
Ala	0	0	0	155	0	0	230	428	0	0	0	0	0	0	361	244	957	0	0	771
Arg	0	0	0	0	694	782	0	1771	678	37	491	580	140	0	480	445	198	416	0	0
Asn	0	0	0	1112	0	0	0	0	124	222	0	475	0	0	0	639	225	0	98	0
Asp	352	0	412	0	0	0	584	939	121	0	0	0	0	0	0	0	0	0	141	163
Cys	0	1887	0	0	0	0	0	297	0	0	0	0	0	170	0	365	0	226	841	0
Gln	0	2215	0	0	0	0	434	0	335	0	174	207	0	0	177	0	0	0	0	0
Glu	250	0	0	499	0	277	0	763	0	0	0	626	0	0	0	0	0	0	0	190
Gly	363	737	0	572	189	0	541	0	0	0	0	0	0	0	0	451	0	96	0	312
His	0	2013	169	363	0	487	0	0	0	0	125	0	0	0	211	0	0	0	416	0
Ile	0	66	145	0	0	0	0	0	0	0	187	44	396	111	0	192	870	0	0	1312
Leu	0	489	0	0	0	125	0	0	112	172	0	0	178	720	1765	662	0	77	0	612
Lys	0	418	500	0	0	223	1698	0	0	38	0	0	124	0	0	0	176	0	0	0
Met	0	54	0	0	0	0	0	0	0	351	215	80	0	0	0	0	977	0	0	1173
Phe	0	0	0	0	252	0	0	0	0	193	734	0	0	0	0	577	0	0	118	240
Pro	562	558	0	0	0	262	0	0	187	0	1585	0	0	0	0	658	349	0	0	0
Ser	445	437	986	0	337	0	0	1202	0	135	281	0	0	407	1113	0	392	97	140	0
Thr	1926	167	169	0	0	0	0	0	0	903	0	195	524	0	387	390	0	0	0	0
Trp	0	1632	0	0	173	0	0	126	0	0	69	0	0	0	0	62	0	0	0	0
Tyr	0	0	110	359	697	0	0	0	460	0	0	0	0	93	0	198	0	0	0	0
Val	1662	0	0	288	0	0	192	701	0	1050	673	0	667	152	0	0	0	0	0	0

b)

reference / alternative	Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	Ile	Leu	Lys	Met	Phe	Pro	Ser	Ter	Thr	Trp	Tyr	Val
Ala	0	0	0	7933	0	0	13684	24130	0	0	0	0	0	0	31163	11905	0	50414	0	0	38146
Arg	0	0	0	0	16211	39044	0	75196	34045	1437	15034	42407	5199	0	30105	30291	875	9731	10044	0	0
Asn	0	0	0	62778	0	0	0	0	7233	7487	0	28332	0	0	0	37846	0	15537	0	3969	0
Asp	14837	0	20076	0	0	0	39193	36427	6215	0	0	0	0	0	0	0	0	0	0	3500	4303
Cys	0	89482	0	0	0	0	0	11136	0	0	0	0	0	7093	0	25329	364	0	8118	40242	0
Gln	0	115786	0	0	0	0	28221	0	19182	0	6767	12813	0	0	9447	0	861	0	0	0	0
Glu	10853	0	0	33513	0	20719	0	32539	0	0	0	34311	0	0	0	0	257	0	0	0	5059
Gly	25840	39434	0	33416	6234	0	32069	0	0	0	0	0	0	0	0	27316	247	0	2778	0	14193
His	0	93083	9184	18216	0	30156	0	0	0	0	4801	0	0	0	12550	0	0	0	0	19068	0
Ile	0	3731	6812	0	0	0	0	0	0	0	12471	2917	30041	4949	0	10099	0	60459	0	0	68940
Leu	0	26646	0	0	0	7730	0	0	7035	13158	0	0	13533	37907	104886	38453	363	0	2793	0	44996
Lys	0	23138	24296	0	0	14754	91778	0	0	1509	0	0	4976	0	0	0	217	9007	0	0	0
Met	0	2521	0	0	0	0	0	0	0	23918	11955	3971	0	0	0	0	0	43377	0	0	63238
Phe	0	0	0	0	8185	0	0	0	0	10222	52261	0	0	0	0	33871	0	0	0	7150	12640
Pro	20660	17908	0	0	0	11716	0	0	8385	0	41097	0	0	0	0	25056	0	13778	0	0	0
Ser	37159	24593	57129	0	11318	0	0	54590	0	5384	10472	0	0	13358	81378	0	510	28312	2613	5258	0
Ter	0	29072	0	0	4308	26510	12401	2269	0	0	2713	4030	0	0	0	6504	0	0	14618	11751	0
Thr	112260	10192	8068	0	0	0	0	0	0	46206	0	12918	27884	0	30140	25217	0	0	0	0	0
Trp	0	75848	0	0	5070	0	0	5942	0	0	3312	0	0	0	0	3298	737	0	0	0	0
Tyr	0	0	5157	17445	22390	0	0	0	30213	0	0	0	0	4962	0	11024	549	0	0	0	0
Val	104807	0	0	15310	0	0	10636	29246	0	70330	46822	0	38022	7316	0	0	0	0	0	0	0

Extended data 1: Tables of counts for maps of amino acid substitutions, **a)** Uniprot1 **b)** Uniprot2.


a)
Color Key
-2 -1 0 1 2
Value
Uniprot1 - Disease causing substitutions (normalized)
neg - negative
pho - hydrophobic
pol - polar
pos - positive
neg Asp
neg Glu
pho Ala
pho Gly
pho Ile
pho Leu
pho Met
pho Phe
pho Pro
pho Trp
pho Val
pol Asn
pol Cys
pol Gln
pol Ser
pol Thr
pol Tyr
pos Arg
pos His
pos Lys
reference + type
alternative + type


b)
Color Key
-2 0 2
Value
Uniprot2 - Disease causing substitutions (normalized)
neg - negative
pho - hydrophobic
pol - polar
pos - positive
neg Asp
neg Glu
pho Ala
pho Gly
pho Ile
pho Leu
pho Met
pho Phe
pho Pro
pho Trp
pho Val
pol Asn
pol Cys
pol Gln
pol Ser
pol Thr
pol Tyr
pos Arg
pos His
pos Lys
reference + type
alternative + type

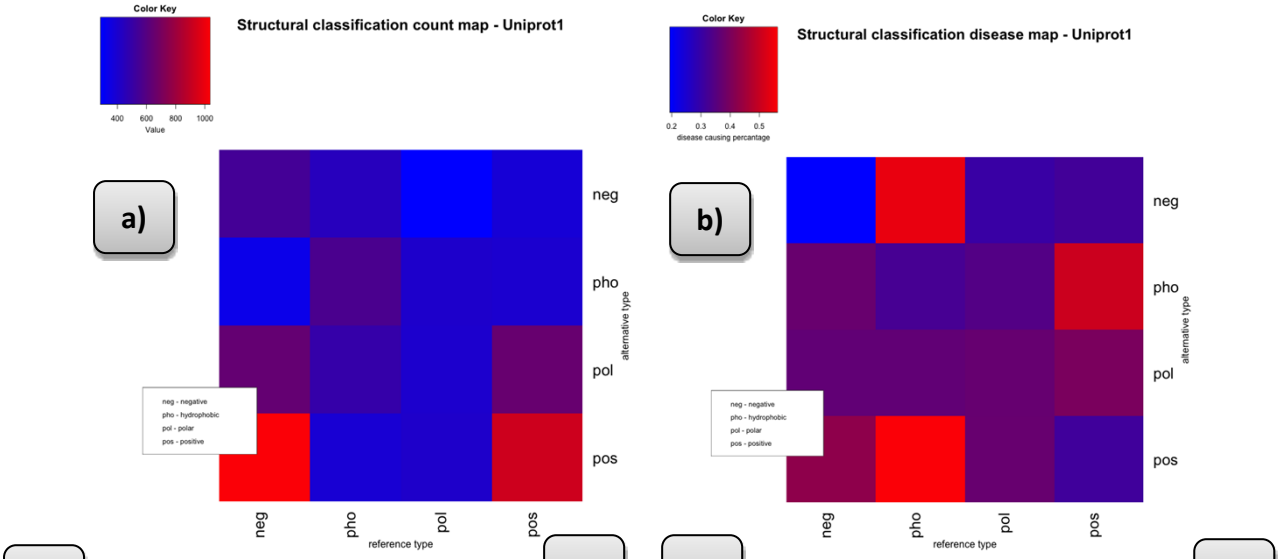
c)

reference / alternative	neg Asp	neg Glu	pho Ala	pho Gly	pho Ile	pho Leu	pho Met	pho Phe	pho Pro	pho Trp	pho Val	pol Asn	pol Cys	pol Gln	pol Ser	pol Thr	pol Tyr	pos Arg	pos His	pos Lys
neg Asp		0.18	0.5	0.58							0.6	0.27					0.64		0.36	
neg Glu	0.21		0.44	0.54							0.52			0.19						0.31
pho Ala	0.32	0.28		0.33					0.2		0.18				0.07	0.15				
pho Gly	0.45	0.29	0.17							0.62	0.46		0.68		0.16			0.41		
pho Ile						0.14	0.28	0.45			0.11	0.56			0.35	0.32		0.23		0.48
pho Leu					0.21		0.2	0.37	0.37	0.52	0.23			0.24	0.4			0.5	0.4	
pho Met					0.28	0.17					0.29					0.31		0.28		0.22
pho Phe					0.5	0.32					0.53		0.75		0.41		0.19			
pho Pro			0.49			0.65								0.51	0.39	0.38		0.65	0.53	
pho Trp				0.52		0.39							0.66		0.56			0.54		
pho Val	0.53	0.33	0.29	0.62	0.08	0.21	0.26	0.55												
pol Asn	0.38				0.63										0.2	0.32	0.57		0.27	0.28
pol Cys				0.65				0.58		0.66					0.27		0.6	0.53		
pol Gln		0.21				0.51			0.31									0.39	0.27	0.2
pol Ser			0.18	0.43	0.61	0.4		0.54	0.26	0.72		0.26	0.53			0.11	0.46	0.41		
pol Thr			0.25		0.37		0.32		0.29			0.31			0.13			0.37		0.3
pol Tyr	0.51							0.16				0.44	0.67		0.35				0.33	
pos Arg				0.6	0.59	0.61	0.67		0.43	0.58			0.64	0.22	0.39	0.49			0.29	0.17
pos His	0.42					0.46			0.32			0.3		0.22			0.39	0.4		
pos Lys		0.43			0.66		0.62					0.4		0.28		0.37		0.14		

d)

reference / alternative	neg-Asp	neg-Glu	pho-Ala	pho-Gly	pho-Ile	pho-Leu	pho-Met	pho-Phe	pho-Pro	pho-Trp	pho-Val	pol-Asn	pol-Cys	pol-Gln	pol-Ser	pol-Thr	pol-Tyr	pos-Arg	pos-His	pos-Lys
neg-Asp		0.0018	0.0049	0.01							0.0114	0.0033					0.0109		0.0056	
neg-Glu	0.0011		0.0075	0.0077							0.0089			0.0046						0.0039
pho-Ala	0.0039	0.0023		0.0047					0.0019		0.0017				0.0014	0.0017				
pho-Gly	0.004	0.0026	0.0025							0.0115	0.0049		0.0075		0.0014			0.0089		
pho-Ile						0.0008	0.0019	0.0051			0.0015	0.0066			0.003	0.003		0.0032		0.0051
pho-Leu					0.0015		0.0019	0.0038	0.0036	0.0107	0.0026			0.0026	0.0053			0.009	0.0037	
pho-Met					0.0019	0.0014					0.0031					0.0044		0.004		0.0028
pho-Phe					0.004	0.0022					0.0052		0.0089		0.0037		0.0021			
pho-Pro			0.0064			0.0115								0.0055	0.0052	0.0046		0.0133	0.0051	
pho-Trp				0.0081		0.0063							0.0093		0.0097			0.0073		
pho-Val	0.0056	0.004	0.0026	0.0106	0.001	0.0018	0.0026	0.0052												
pol-Asn	0.0036				0.0092										0.0017	0.0021	0.0081		0.0036	0.0026
pol-Cys				0.009				0.0073		0.0111					0.0023		0.0054	0.0068		
pol-Gln		0.0034				0.0068			0.0046									0.0049	0.0023	0.0036
pol-Ser			0.0022	0.0061	0.0067	0.005		0.0076	0.002	0.0165		0.0026	0.0072			0.001	0.008	0.005		
pol-Thr			0.0025		0.0034		0.003		0.0024			0.004			0.0015			0.0034		0.002
pol-Tyr	0.006							0.005				0.0043	0.0084		0.0039				0.0026	
pos-Arg				0.0088	0.0063	0.0105	0.0125		0.0034	0.0124			0.011	0.0022	0.0024	0.0053			0.003	0.0014
pos-His	0.0046					0.0069			0.0033			0.0032		0.0017			0.0047	0.0063		
pos-Lys		0.0057			0.0106		0.0084					0.0037		0.0061		0.0047		0.0016		

Extended data 2: **a)** normalized Uniprot1 map of disease casing variants, **b)** normalized Uniprot2 map of disease casing variants, **c)** table for Uniprot1 map of disease causing variants, **d)** table for Uniprot2 map of disease causing variants.



reference / alternative	neg	pho	pol	pos
neg	542	444	283	380
pho	332	565	405	398
pol	641	490	400	651
pos	1036	385	410	922

reference / alternative	neg	pho	pol	pos
neg	0.19	0.54	0.3	0.32
pho	0.37	0.32	0.34	0.51
pol	0.36	0.36	0.37	0.4
pos	0.43	0.56	0.37	0.31

reference / alternative	neg	pho	pol	pos
neg	77880	35196	31222	42593
pho	40123	65883	43004	39496
pol	76740	54683	44441	74183
pos	116194	32812	42045	102340

reference / alternative	neg	pho	pol	pos
neg	0.0007	0.0041	0.0021	0.002
pho	0.0017	0.0016	0.0018	0.0038
pol	0.0019	0.0019	0.0017	0.0022
pos	0.0026	0.0038	0.0019	0.0019

Extended data 3: Maps based on structural classification, **a)** Uniprot1 map of normalized counts, **b)** Uniprot1 map of disease percentages, **c,d)** tables for Uniprot1 maps, **e,f)** tables for Uniprot2 maps.

a)

reference / alternative	Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	Ile	Leu	lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val
Ala	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	4	0	0	0
Arg	0	4	0	0	2	0	0	6	0	0	0	0	0	0	0	2	0	2	0	0
Asn	0	0	0	2	0	0	0	0	2	0	0	0	0	0	0	0	0	0	2	0
Asp	0	0	2	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	2	0
Cys	0	2	0	0	0	0	0	2	0	0	0	0	0	0	0	2	0	0	0	0
Gln	0	0	0	0	0	0	2	0	0	0	0	2	0	0	0	0	0	0	0	0
Glu	0	0	0	0	0	2	0	0	0	0	0	2	0	0	0	0	0	0	0	0
Gly	0	6	0	0	2	0	0	0	0	0	0	0	0	0	0	2	0	1	0	0
His	0	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0
Ile	0	0	0	0	0	0	0	0	0	0	4	0	0	2	0	0	0	0	0	3
Leu	0	0	0	0	0	0	0	0	0	4	4	0	2	2	0	0	0	0	0	6
lys	0	0	0	0	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0
Met	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	1
Phe	0	0	0	0	0	0	0	0	0	2	2	0	0	0	0	0	0	0	0	2
Pro	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	0	0	0
Ser	4	2	0	0	2	0	0	2	0	0	0	0	0	0	4	0	4	0	0	0
Thr	4	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	0	0	0	0
Trp	0	2	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Tyr	0	0	2	2	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0
Val	0	0	0	0	0	0	0	0	0	3	6	0	1	2	0	0	0	0	0	0

b)

reference / alternative	Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	Ile	Leu	lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val
Ala	0	0	0	2	0	0	2	4	0	0	0	0	0	0	0	0	0	0	0	4
Arg	0	0	0	0	0	2	0	0	2	1	4	2	1	0	4	0	2	0	0	0
Asn	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	2	2	0	0	0
Asp	2	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	2
Cys	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	2	0	0	2	0
Gln	0	2	0	0	0	0	0	0	0	0	2	0	0	0	2	0	0	0	0	0
Glu	2	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	2
Gly	4	0	0	2	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	4
His	0	2	0	0	0	0	0	0	0	0	2	0	0	0	2	0	0	0	0	0
Ile	0	1	2	0	0	0	0	0	0	0	0	1	0	0	0	2	3	0	0	0
Leu	0	4	0	0	0	2	0	0	2	0	0	0	0	0	4	2	0	1	0	0
lys	0	2	0	0	0	0	0	0	0	1	0	0	1	0	0	0	2	0	0	0
Met	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0
Phe	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	2	0	0	2	0
Pro	0	4	0	0	0	2	0	0	2	0	4	0	0	0	0	0	0	0	0	0
Ser	0	0	2	0	2	0	0	0	0	2	2	0	0	2	0	0	2	1	2	0
Thr	0	2	2	0	0	0	0	0	0	3	0	2	1	0	0	2	0	0	0	0
Trp	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0
Tyr	0	0	0	0	2	0	0	0	0	0	0	0	0	2	0	2	0	0	0	0
Val	4	0	0	2	0	0	2	4	0	0	0	0	0	0	0	0	0	0	0	0

c)

reference / alternative	Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	Ile	Leu	lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val
Ala	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Arg	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0
Asn	0	0	2	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0
Asp	0	0	0	2	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0
Cys	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0
Gln	0	0	0	0	0	2	0	0	4	0	0	0	0	0	0	0	0	0	0	0
Glu	0	0	0	4	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
Gly	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0
His	0	0	0	0	0	4	0	0	2	0	0	0	0	0	0	0	0	0	0	0
Ile	0	0	0	0	0	0	0	0	0	6	0	0	3	0	0	0	0	0	0	0
Leu	0	0	0	0	0	0	0	0	0	0	14	0	0	4	0	0	0	0	0	0
lys	0	0	4	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0
Met	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0
Phe	0	0	0	0	0	0	0	0	0	0	4	0	0	2	0	0	0	0	0	0
Pro	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0
Ser	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	14	0	0	0	0
Thr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0
Trp	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Tyr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0
Val	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12

Extended data 4: Frequencies of amino acid substitutions based on the genetic code table, **a)** 1st codon base substitutions, **b)** 2nd codon base substitutions, **c)** 3rd codon base substitutions.

a)

reference / alternative	Ala	Cys	Asp	Glu	Phe	Gly	His	Ile	lys	Leu	Met	Asn	Pro	Gln	Arg	Ser	Thr	Val	Trp	Tyr
Ala			0.22	0.18		0.2							0.18			0.16	0.17	0.18		
Cys					0.2	0.17									0.18	0.17			0.17	0.18
Asp	0.21			0.17		0.18	0.16					0.18						0.18		0.18
Glu	0.17		0.17			0.17			0.18					0.17				0.12		
Phe		0.19						0.2		0.19						0.17		0.18		0.17
Gly	0.17	0.21	0.18	0.2											0.15	0.17		0.22	0.24	
His			0.18							0.15		0.21	0.17	0.19	0.18					0.19
Ile					0.2				0.18	0.16	0.19	0.19			0.25	0.15	0.16	0.17		
lys				0.18				0.29			0.18	0.17		0.19	0.18		0.19			
Leu					0.18		0.17	0.17			0.17		0.17	0.16	0.18	0.17		0.17	0.14	
Met								0.2	0.13	0.18					0.12		0.17	0.18		
Asn			0.18				0.15	0.18	0.18							0.18	0.19			0.18
Pro	0.19						0.19			0.16				0.14	0.19	0.18	0.2			
Gln				0.18			0.17		0.17	0.18			0.17		0.18					
Arg		0.18				0.19	0.18	0.16	0.18	0.17	0.16		0.19	0.18		0.19	0.15		0.17	
Ser	0.17	0.2			0.19	0.18		0.21		0.16		0.18	0.18		0.18		0.18		0.17	0.22
Thr	0.17							0.18	0.2		0.17	0.19	0.18		0.18	0.19				
Val	0.18		0.21	0.18	0.18	0.2		0.17		0.17	0.16									
Trp		0.24				0.18				0.2					0.18	0.13				
Tyr		0.17	0.18		0.18		0.17					0.13				0.14				

b)

reference / alternative	Ala	Cys	Asp	Glu	Phe	Gly	His	Ile	lys	Leu	Met	Asn	Pro	Gln	Arg	Ser	Thr	Val	Trp	Tyr
Ala			0.19	0.19		0.18							0.19			0.19	0.18	0.19		
Cys					0.18	0.18									0.18	0.19			0.17	0.19
Asp	0.18			0.19		0.18	0.19					0.19						0.18		0.19
Glu	0.2		0.19			0.2			0.18					0.2				0.19		
Phe		0.19						0.17		0.19						0.17		0.19		0.18
Gly	0.18	0.21	0.19	0.2											0.19	0.18		0.2	0.17	
His			0.18							0.2		0.19	0.18	0.18	0.19					0.19
Ile					0.18				0.17	0.19	0.19	0.2			0.19	0.21	0.19	0.19		
lys				0.18				0.17			0.19	0.18		0.21	0.19		0.2			
Leu					0.2		0.16	0.19			0.19		0.18	0.18	0.18	0.19		0.19	0.18	
Met								0.19	0.13	0.2					0.16		0.19	0.19		
Asn			0.19				0.2	0.21	0.18							0.18	0.18			0.16
Pro	0.19						0.2			0.18				0.19	0.2	0.19	0.21			
Gln				0.18			0.18		0.19	0.18			0.18		0.19					
Arg		0.19				0.19	0.19	0.22	0.19	0.19	0.19		0.19	0.18		0.19	0.16		0.18	
Ser	0.19	0.19			0.18	0.19		0.17		0.2		0.19	0.19		0.19		0.19		0.2	0.19
Thr	0.19							0.18	0.2		0.19	0.2	0.2		0.18	0.18				
Val	0.19		0.15	0.21	0.17	0.18		0.19		0.2	0.18									
Trp		0.18				0.17				0.18					0.18	0.18				
Tyr		0.19	0.18		0.19		0.18					0.19				0.18				

c)

reference / alternative	Ala	Cys	Asp	Glu	Phe	Gly	His	Ile	lys	Leu	Met	Asn	Pro	Gln	Arg	Ser	Thr	Val	Trp	Tyr
Ala			0.25	0.26		0.26							0.26			0.3	0.28	0.28		
Cys					0.29	0.29									0.28	0.26			0.32	0.27
Asp	0.26			0.28		0.25	0.23					0.27						0.26		0.29
Glu	0.25		0.27			0.26			0.29					0.25				0.35		
Phe		0.25						0.3		0.26						0.3		0.29		0.24
Gly	0.29	0.25	0.26	0.24											0.28	0.3		0.23	0.27	
His			0.28							0.22		0.28	0.26	0.27	0.26					0.27
Ile					0.24				0.25	0.27	0.28	0.22			0.21	0.3	0.3	0.28		
lys				0.27				0.21			0.26	0.31		0.26	0.26		0.25			
Leu					0.27		0.32	0.29			0.24		0.27	0.31	0.26	0.28		0.28	0.3	
Met								0.26	0.46	0.23					0.38		0.25	0.27		
Asn			0.25				0.27	0.23	0.26							0.29	0.3			0.31
Pro	0.28						0.28			0.3				0.28	0.24	0.27	0.23			
Gln				0.27			0.27		0.26	0.27			0.28		0.26					
Arg		0.28				0.26	0.28	0.23	0.28	0.27	0.25		0.26	0.28		0.28	0.31		0.29	
Ser	0.26	0.26			0.29	0.26		0.3		0.26		0.28	0.26		0.27		0.28		0.14	0.26
Thr	0.27							0.28	0.23		0.29	0.25	0.26		0.24	0.28				
Val	0.26		0.33	0.21	0.28	0.27		0.27		0.27	0.29									
Trp		0.24				0.27				0.29					0.27	0.38				
Tyr		0.28	0.27		0.28		0.29					0.31				0.33				

d)

reference / alternative	Ala	Cys	Asp	Glu	Phe	Gly	His	Ile	lys	Leu	Met	Asn	Pro	Gln	Arg	Ser	Thr	Val	Trp	Tyr
Ala			0.19	0.18		0.18							0.18			0.17	0.18	0.18		
Cys					0.14	0.17									0.18	0.19			0.17	0.19
Asp	0.17			0.17		0.19	0.22					0.19						0.2		0.18
Glu	0.19		0.19			0.19			0.17					0.19				0.18		
Phe		0.17						0.16		0.19						0.17		0.18		0.21
Gly	0.19	0.15	0.2	0.19											0.19	0.18		0.2	0.14	
His			0.19							0.22		0.14	0.19	0.18	0.18					0.18
Ile					0.2				0.2	0.19	0.17	0.2			0.17	0.18	0.19	0.18		
lys				0.18				0.18			0.18	0.17		0.18	0.18		0.17			
Leu					0.18		0.18	0.18			0.19		0.18	0.16	0.19	0.18		0.18	0.22	
Met								0.17	0.14	0.21					0.19		0.19	0.18		
Asn			0.19				0.2	0.2	0.18							0.17	0.17			0.15
Pro	0.16						0.18			0.18				0.19	0.2	0.18	0.21			
Gln				0.18			0.19		0.18	0.19			0.17		0.19					
Arg		0.18				0.18	0.17	0.22	0.17	0.18	0.18		0.18	0.18		0.16	0.18		0.17	
Ser	0.18	0.18			0.16	0.18		0.16		0.19		0.17	0.18		0.18		0.18		0.27	0.15
Thr	0.18							0.18	0.19		0.18	0.21	0.19		0.21	0.17				
Val	0.18		0.14	0.2	0.19	0.17		0.18		0.19	0.19									
Trp		0.19				0.19				0.15					0.18	0.14				
Tyr		0.18	0.18		0.18		0.18					0.19				0.18				

e)

reference / alternative	Ala	Cys	Asp	Glu	Phe	Gly	His	Ile	lys	Leu	Met	Asn	Pro	Gln	Arg	Ser	Thr	Val	Trp	Tyr
Ala			0.16	0.19		0.18							0.19			0.18	0.18	0.18		
Cys					0.19	0.2									0.19	0.18			0.17	0.18
Asp	0.18			0.2		0.19	0.2					0.18						0.18		0.17
Glu	0.19		0.19			0.18			0.18					0.19				0.16		
Phe		0.19						0.16		0.18						0.18		0.16		0.2
Gly	0.18	0.18	0.18	0.17											0.19	0.18		0.16	0.18	
His			0.17							0.21		0.18	0.21	0.19	0.19					0.19
Ile					0.18				0.2	0.18	0.18	0.19			0.18	0.17	0.17	0.18		
lys				0.19				0.15			0.19	0.17		0.17	0.19		0.18			
Leu					0.17		0.17	0.17			0.2		0.19	0.19	0.19	0.18		0.18	0.16	
Met								0.18	0.14	0.18					0.16		0.19	0.18		
Asn			0.19				0.18	0.17	0.2							0.18	0.15			0.19
Pro	0.17						0.16			0.18				0.2	0.17	0.17	0.15			
Gln				0.2			0.18		0.2	0.18			0.19		0.19					
Arg		0.18				0.18	0.17	0.17	0.18	0.19	0.21		0.18	0.18		0.18	0.2		0.2	
Ser	0.2	0.19			0.18	0.19		0.16		0.19		0.18	0.18		0.18		0.18		0.22	0.17
Thr	0.19							0.18	0.19		0.18	0.15	0.18		0.2	0.18				
Val	0.18		0.17	0.19	0.17	0.18		0.18		0.18	0.18									
Trp		0.15				0.18				0.18					0.19	0.17				
Tyr		0.19	0.18		0.16		0.18					0.18				0.17				

Extended data 5: Table of percentages for maps of frequency resolved amino acid substitutions in each population group. **a)** East Asian, **b)** American, **c)** African, **d)** European, **e)** South Asian

§ 8. Literature

1. D. Vitkup, C. Sander, G.M. Church, *The amino-acid mutational spectrum of human genetic disease, Genome Biol.* **4** (2003) R72.
2. H.J.C. Berendsen, et al., *GROMACS: A message-passing parallel molecular dynamics implementation*, *Comp. Phys. Comm*. **91 (**1995) 43–56.
3. I. Wagner, H. Musso, *New Naturally Occurring Amino Acids, Angew. Chem. Int. Ed. Engl.* **22** (1983) 816–828.
4. D.M. Driscoll, P.R. Copeland, *Mechanism and regulation of selenoprotein synthesis, Annu Rev Nutr.* **23** (2003) 17-40.
5. J.A. Krzycki, *The direct genetic encoding of pyrrolysine, Curr Opin Microbiol.* **8** (2005) 706-712.
6. A. Haoudi, H. Bensmail, *Bioinformatics and data mining in proteomics, Expert Rev Proteomics.* **3** (2006) 333-343.
7. http://www.uniprot.org (21.09.2016.)
8. M.S. Kim, et al., *A draft map of the human proteome*, *Nature.* **509** (2014) 575-581.
9. M. B. Gerstein, et al., *What is a gene, post-ENCODE? History and updated definition, Genome research* **17** (2007) 669-681.
10. A. S. Kauffman, *Metabolic stability and epigenesis in randomly constructed genetic nets, Journal of theoretical biology* **22** (1969) 437-467.
11. ENCODE Project Consortium, *An integrated encyclopedia of DNA elements in the human genome, Nature* **489** (2012) 57-74.
12. E. S. Lander, et al., *Initial sequencing and analysis of the human genome, Nature* **409** (2001) 860-921.
13. https://en.wikipedia.org/wiki/Amino_acid (21.09.2016.)
14. G. D. Rose, et al., *A backbone-based theory of protein folding, Proceedings of the National Academy of Sciences* **103** (2006) 16623-16633.
15. http://www.proteinstructures.com/Structure/Structure/Ramachandran-plot.html (21.09.2016.)
16. P. Edman, *Method for determination of the amino acid sequence in peptides, Acta chem. scand.* **4** (1950) 283-293.
17. R.C. Bi, et al., *Protein crystallization in space, Microgravity Sci Technol.* **7** (1994) 203-206.
18. L. J. DeLucas, et al., *Protein crystal growth and the International Space Station, Gravitational and Space Research* **12** (2007) 39-45.
19. E. Callaway, *The revolution will not be crystallized: a new method sweeps through structural biology, Nature* **525** (2015) 172.
20. D. L. Nelson, M. C. Cox, *Lehninger Principles of Biochemistry*, W.H. Freeman, New York, 2012.
21. S. Freeman, *Biological Science*, Pearson Prentice Hall, Upper Saddle River, 2005.
22. J. M. Berg, J. L. Tymoczko, L. Stryer, *Biochemistry*, W. H. Freeman and Company, New York, 2012.
23. https://www.nlm.nih.gov/medlineplus/ency/article/002222.htm (21.09.2016.)
24. https://en.wikipedia.org/wiki/Transcription_%28genetics%29 (21.09.2016.)
25. http://gtrnadb.ucsc.edu/ (21.09.2016.)
26. F. H. C. Crick, *Codon—anticodon pairing: the wobble hypothesis, Journal of molecular biology* **19** (1966) 548-555.
27. F. V. Murphy, V. Ramakrishnan, *Structure of a purine-purine wobble base pair in the decoding center of the ribosome, Nature structural & molecular biology* **11** (2004) 1251-1252.
28. M. Kimura, The neutral theory of molecular evolution, Cambridge University Press, Cambridge, 1984.
29. J. C. Roach, et al., *Analysis of genetic inheritance in a family quartet by whole-genome sequencing, Science* **328** (2010) 636-639.
30. M. W. Nachman, S. L. Crowell, *Estimate of the mutation rate per nucleotide in humans, Genetics* **156** (2000) 297-304.
31. 1000 Genomes Project Consortium, *A global reference for human genetic variation, Nature* **526** (2015) 68-74.
32. https://en.wikipedia.org/wiki/Transition_%28genetics%29 (21.09.2016.)
33. R. K. Saiki, et al., *Enzymatic amplification of b-globin genomic sequences and restriction site analysis for diagnosis of sickle cell anemia, Science* **230** (1985) 1350-1354.
34. E. E. Schadt, S. Turner, A. Kasarskis, *A window into third-generation sequencing, Human molecular genetics* **19** (2010) R227-R240.

35. F. Sanger, S. Nicklen, A. R. Coulson, *DNA sequencing with chain-terminating inhibitors, Proceedings of the National Academy of Sciences* **74** (1977) 5463-5467.
36. R. Padmanabhan, E. Jay, R. Wu, *Chemical synthesis of a primer and its use in the sequence analysis of the lysozyme gene of bacteriophage T4, Proceedings of the National Academy of Sciences* **71** (1974) 2510-2514.
37. F. Sanger, *The nucleotide sequence of bacteriophage φX174, Journal of Molecular Biology* **125** (1978) 225-246.
38. S. Anderson, *Shotgun DNA sequencing using cloned DNase I-generated fragments, Nucleic Acids Research* **9** (1981) 3015-3027.
39. J. C. Venter, et al., *The sequence of the human genome, science* **291** (2001) 1304-1351.
40. M. J. P. Chaisson, et al., *Resolving the complexity of the human genome using single-molecule sequencing, Nature* **517** (2015) 608-611.
41. J. D. Boeke, et al., *The Genome Project-Write, Science* **353** (2016) 126-127.
42. M. L. Metzker, *Sequencing technologies—the next generation, Nature reviews genetics* **11** (2010) 31-46.
43. M. Ronaghi, M. Uhlén, P. Nyren, *A sequencing method based on real-time pyrophosphate, Science* **281** (1998) 363.
44. J. H. Leamon, et al., *A massively parallel PicoTiterPlate based platform for discrete picoliter‐scale polymerase chain reactions, Electrophoresis* **24** (2003) 3769-3777.
45. D. R. Bentley, et al., *Accurate whole human genome sequencing using reversible terminator chemistry, nature* **456** (2008) 53-59.
46. J. Eid, et al., *Real-time DNA sequencing from single polymerase molecules, Science* **323** (2009) 133-138.
47. M. J. Levene, et al., *Zero-mode waveguides for single-molecule analysis at high concentrations, Science* **299** (2003) 682-686.
48. C.-S. Chin, et al., *Nonhybrid, finished microbial genome assemblies from long-read SMRT sequencing data, Nature methods* **10** (2013) 563-569.
49. O. J. rivanek, et al., *Atom-by-atom structural and chemical analysis by annular dark-field electron microscopy, Nature* **464** (2010) 571-574.
50. https://en.wikipedia.org/wiki/Transmission_electron_microscopy_DNA_sequencing (21.09.2016.)
51. M. Xu, D. Fujita, N. Hanagata, *Perspectives and Challenges of Emerging Single‐Molecule DNA Sequencing Technologies, Small* **5** (2009) 2638-2649.
52. T. Ohshiro, et al., *Single-molecule electrical random resequencing of DNA and RNA, Scientific reports* **2** (2012), 501.
53. https://microscopyinnovations.com/transmission-electron-microscopy-tem/ (21.09.2016.)
54. https://www.nanoporetech.com (21.09.2016.)
55. D. Deamer, M. Akeson, D. Branton, *Three decades of nanopore sequencing, Nature biotechnology* **34** (2016) 518-524.
56. T. Laver, et al., *Assessing the performance of the Oxford Nanopore Technologies MinION, Biomolecular detection and quantification* **3** (2015) 1-8.
57. http://theconversation.com/how-a-small-backpack-for-fast-genomic-sequencing-is-helping-combat-ebola-41863 (21.09.2016.)
58. M. Jain, et al., *Improved data analysis for the MinION nanopore sequencer, Nature methods* **12** (2015) 351-356.
59. http://www.1000genomes.org/ (21.09.2016.)
60. L. B. Barreiro, et al., *Natural selection has driven population differentiation in modern humans, Nature genetics* **40** (2008) 340-345.
61. M. N. Weedon, et al., *Genome-wide association analysis identifies 20 loci that influence adult height, Nature genetics* **40** (2008) 575-583.
62. P. R. Burton, et al., *Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls, Nature* **447** (2007) 661-678.
63. E. A. Repnikova, et al., *Characterization of copy number variation in genomic regions containing STR loci using array comparative genomic hybridization, Forensic Science International: Genetics* **7** (2013) 475-481.
64. 1000 Genomes Project Consortium, *A map of human genome variation from population-scale sequencing, Nature* **467** (2010) 1061-1073.

65. P. H. Sudmant, et al., *An integrated map of structural variation in 2,504 human genomes, Nature* **526** (2015) 75-81.
66. A. M. Mood, *Introduction to the Theory of Statistics*, McGraw Hill, New York, 1974.
67. A. Loy, L. Follett, H. Hofmann, *Variations of QQ Plots–the Power of our Eyes!, The American Statistician* **accepted** (2015)
68. C. Trapnell, S. L. Salzberg, *How to map billions of short reads onto genomes, Nature biotechnology* **27** (2009) 455.
69. M. Pop, *Genome assembly reborn: recent computational challenges, Briefings in bioinformatics* **10** (2009) 354-366.
70. A. Carvajal-Rodríguez, *Simulation of genes and genomes forward in time, Current genomics* **11** (2010) 58-61.
71. T. S. Furey, *ChIP–seq and beyond: new and improved methodologies to detect and characterize protein–DNA interactions, Nature Reviews Genetics* **13** (2012) 840-852.
72. L. Song, G. E. Crawford, *DNase-seq: a high-resolution technique for mapping active gene regulatory elements across the genome from mammalian cells, Cold Spring Harbor Protocols* **2010** (2010) 1-12.
73. B. Van Steensel, J. Dekker, *Genomics tools for unraveling chromosome architecture, Nature biotechnology* **28** (2010) 1089-1095.
74. https://www.genome.gov/27541954/dna-sequencing-costs-data/ (21.09.2016.)
75. http://www.ncbi.nlm.nih.gov/genbank/statistics/ (21.09.2016.)
76. J. L. Klepeis, et al., *Long-timescale molecular dynamics simulations of protein structure and function, Current opinion in structural biology* **19** (2009) 120-127.
77. B. Bertoša, et al., *Homooligomerization is needed for stability: a molecular modelling and solution study of Escherichia coli purine nucleoside phosphorylase, FEBS Journal* **281** (2014) 1860-1871.
78. M. U. Johansson, et al., *Defining and searching for structural motifs using DeepView/Swiss-PdbViewer, BMC bioinformatics* **13** (2012) 1.
79. K. Lindorff-Larsen, et al., *Improved side‐chain torsion potentials for the Amber ff99SB protein force field, Proteins: Structure, Function, and Bioinformatics* **78** (2010) 1950-1958.
80. ftp://ftp.uniprot.org/pub/databases/uniprot/previous_releases/release-2016_02/knowledgebase/knowledgebase2016_02.tar.gz (21.09.2016.)
81. ftp://ftp.uniprot.org/pub/databases/uniprot/current_release/knowledgebase/variants/ (21.09.2016.)
82. ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/release/20130502/supporting/functional_annotation/filtered/ (21.09.2016.)
83. http://ensembl.org/info/genome/variation/data_description.html (21.09.2016.)
84. I. A. Adzhubei, et al., *A method and server for predicting damaging missense mutations, Nature methods* **7** (2010) 248-249.
85. P. Kumar, S. Henikoff, P. C. Ng, *Predicting the effects of coding non-synonymous variants on protein function using the SIFT algorithm, Nat Protoc.* **4** (2009) 1073-1081.

§ 9. Supplements

9.1. Code for programming language R

Most of the analyses for this thesis were conducted in programming language R. The computational code is ready for execution and provided with the electronic version of the thesis on the accompanying DVD, together with all required datasets. It's also pasted here for quick reference.

```
##############################################################################################
# Uniprot1
##############################################################################################
library("seqinr")
library("dplyr")
library("Biostrings")
library("tidyr")
library("gplots")
color.final <- colorRampPalette(c("blue","red"), space = "rgb")(299)

# loading and extracting data
# !!setwd to location of datasets
load <- read.fwf("uniprot1.txt", widths = c(10, 11, 12, 15, 14, 12, 10000), header = F, skip
= 30, stringsAsFactors = F)
old <- substr(load[,4], 3, 5)
new <- sub(".{5}[[:digit:]]*", "", load[,4])
new <- substr(new, 1, 3)
dis <- ifelse(load$V7 == "-", F, T)
sub <- data.frame(old, new)
sub <- sub[1:(nrow(sub) - 5),]
res <- table(sub)[,4:23]
res <- res[-c(1,15,18,23),]
res <- t(res)

# Genetic code
g <- GENETIC_CODE
t0 <- names(g)
codeSwitch <- function(code, position, depth){
  if(substr(code, position, position) == "T"){
    substr(code, position, position) <- "G"
  } else if(substr(code, position, position) == "G"){
    substr(code, position, position) <- "C"
  } else if(substr(code, position, position) == "C"){
    substr(code, position, position) <- "A"
  } else if(substr(code, position, position) == "A"){
    substr(code, position, position) <- "T"
  }
  if(depth != 1){
    return(codeSwitch(code, position, depth - 1))
  } else {
    return(code)
  }
}
for(i in 1:3)
  for(j in 1:3)
    assign(paste("t",i,j, sep = ""), sapply(t0, codeSwitch, position = i, depth = j))
codes <- data.frame(t0,t11,t12,t13,t21,t22,t23,t31,t32,t33)
codes.aa <- data.frame(g[t0])
for(i in 2:10)
  codes.aa[,i] <- g[as.vector(codes[,i])]
codes.aa.res <- table(codes.aa[[1]], codes.aa[[2]])
for(i in 3:10)
  codes.aa.res <- codes.aa.res + table(codes.aa[[1]], codes.aa[[i]])
codes.aa.res <- codes.aa.res[-1,-1]
```

```
rownames(codes.aa.res) <-
c("Ala","Cys","Asp","Glu","Phe","Gly","His","Ile","lys","Leu","Met","Asn","Pro","Gln","Arg","
Ser","Thr","Val","Trp","Tyr")
colnames(codes.aa.res) <-
c("Ala","Cys","Asp","Glu","Phe","Gly","His","Ile","lys","Leu","Met","Asn","Pro","Gln","Arg","
Ser","Thr","Val","Trp","Tyr")
codes.aa.res <- t(codes.aa.res)
codes.aa.res <- codes.aa.res[order(rownames(codes.aa.res)), order(colnames(codes.aa.res))]

# counting by AA type
res.x <- res
res.x[res.x < 10] <- NA
heatmap.2(res.x, Rowv = F, Colv = "Rowv", col = color.final, dendrogram = "none", trace =
"none", density.info = "none", xlab = "reference", ylab = "alternative",
          key.xlab = "count", main = "Uniprot1 map")

# normalization
type <-
c("pho","pos","pol","neg","pol","pol","neg","pho","pos","pho","pho","pos","pho","pho","pho","
pol","pol","pho","pol","pho")
res.type2 <- rowsum(t(rowsum(t(res), type)), type)
codes.aa.res.type <- codes.aa.res
diag(codes.aa.res.type) <- 0
codes.aa.res.type[codes.aa.res.type > 0] <- 1
codes.aa.res.type <- rowsum(t(rowsum(t(codes.aa.res.type), type)), type)
codes.aa.res.type.full <- rowsum(t(rowsum(t(matrix(1, 20, 20)), type)), type)
heatmap.2(res.type2/codes.aa.res.type, Rowv = F, Colv = "Rowv", col = color.final, dendrogram
= "none", trace = "none", density.info = "none", xlab = "reference type",
          ylab = "alternative type",main = "Structural classification count map - Uniprot1")
legend("bottomleft",c("neg - negative","pho - hydrophobic","pol - polar","pos - positive"),
cex = 0.7)

diss <- subset(load, V5 == "Disease       ")
nondiss <- subset(load, V5 == "Polymorphism  ")
old.diss<- substr(diss[,4], 3, 5)
new.diss <- sub(".{5}[[:digit:]]*", "", diss[,4])
new.diss <- substr(new.diss, 1, 3)
sub.diss <- data.frame(old.diss, new.diss)
res.diss <- table(sub.diss)[-16,]
res.diss <- t(res.diss)
old.nondiss<- substr(nondiss[,4], 3, 5)
new.nondiss <- sub(".{5}[[:digit:]]*", "", nondiss[,4])
new.nondiss <- substr(new.nondiss, 1, 3)
sub.nondiss <- data.frame(old.nondiss, new.nondiss)
res.nondiss <- table(sub.nondiss)
res.nondiss <- t(res.nondiss)
res.type.diss <- rowsum(res.diss, type)
res.type.diss <- res.type.diss/as.vector(table(type))
res.type.diss <- t(rowsum(t(res.type.diss), type))
res.type.norm.count.diss <- t(t(res.type.diss) / as.vector(table(type)))
res.type.norm.count.perc.diss <- res.type.norm.count.diss/sum(res.type.norm.count.diss)
res.type.nondiss <- rowsum(res.nondiss, type)
res.type.nondiss <- res.type.nondiss/as.vector(table(type))
res.type.nondiss <- t(rowsum(t(res.type.nondiss), type))
res.type.norm.count.nondiss <- t(t(res.type.nondiss) / as.vector(table(type)))
res.type.norm.count.perc.nondiss <-
res.type.norm.count.nondiss/sum(res.type.norm.count.nondiss)

# type analysis
res.type.diss <- rowsum(t(rowsum(t(res.diss), type)), type)
res.type.diss <- res.type.diss/res.type2
heatmap.2(res.type.diss, Rowv = F, Colv = "Rowv", col = color.final, dendrogram = "none",
trace = "none", density.info = "none", xlab = "reference type", ylab =
            "alternative type", key.xlab = "disease causing percantage",main = "Structural
classification disease map - Uniprot1")
legend("bottomleft",c("neg - negative","pho - hydrophobic","pol - polar","pos - positive"),
cex = 0.7)

# AA that change the most
proteins.used <- unique(load$V2[1:73234])
proteins.used <- substr(proteins.used, 1, 6)
proteins.used[proteins.used == "A0A087"] <- "XXXXXX"
ss <- read.fasta("supplement.fasta", seqtype = "AA")
match("sp|A0A087X1C5|CP2D7_HUMAN", names(ss))
names(ss)[3702] <- "XXXXXXXXXXXX"
names(ss) <- substr(names(ss), 4, 9)
ss <- ss[names(ss) %in% proteins.used]
```

```
all <- unname(unlist(ss))
aa.data <- colSums(res)
aa.all <- table(all)
names(aa.all) <-
c("Ala","Cys","Asp","Glu","Phe","Gly","His","Ile","lys","Leu","Met","Asn","Pro","Gln","Arg","
Ser","Thr","xxx","Val","Trp","xxx","Tyr")
aa.all <- aa.all[order(names(aa.all))][-(21:22)]
par(bg = "white")
plot(aa.data/aa.all, xaxt = "n")
axis(1, at = 1:20, labels = names(aa.all))
barplot(aa.data/aa.all, xlab = "residue", ylab = "pecentage of amino acids affected by
variants", col = "black", ylim = c(0,0.025))

barplot(colSums(res.diss)/(colSums(res)), xlab = "reference amino acid", ylab = "pecentage of
disease causing substitutions", ylim = c(0,0.7), main = "Uniprot1")
barplot(rowSums(res.diss)/rowSums(res), xlab = "alternative amino acid", ylab = "pecentage of
disease causing substitutions", ylim = c(0,0.6), main = "Uniprot1")

area <-
matrix(c(0.5,20.5,0.5,18.5,2.5,18.5,2.5,9.5,11.5,9.5,11.5,3.5,17.5,3.5,17.5,0.5,20.5,0.5,20.5
,3.5,17.5,3.5,17.5,9.5,11.5,9.5,11.5,18.5,2.5,18.5,2.5,20.5), ncol
               = 2, byrow = T)
ress <- res
ress[ress < 10] <- 0
res.diss[res.diss < 10] <- 0
aa.dis.subs <- res.diss/ress
rownames(aa.dis.subs) <- paste(type, rownames(aa.dis.subs))
colnames(aa.dis.subs) <- paste(type, colnames(aa.dis.subs))
aa.dis.subs <- aa.dis.subs[order(rownames(aa.dis.subs)), order(colnames(aa.dis.subs))]
aa.dis.subs
heatmap.2(aa.dis.subs, Rowv = NA, Colv = "Rowv", col = color.final, scale = "none", xlab =
"reference + type", ylab = "alternative + type", main = "Uniprot1 - Disease
          causing substitutions",
          add.expr = polygon(area, lwd = 3, border = 3), dendrogram = "none", trace = "none",
density.info = "none", key.xlab = "percentage of disease causing")
legend("bottomleft",c("neg - negative","pho - hydrophobic","pol - polar","pos - positive"),
cex = 0.65)
# normalized:
heatmap.2(replace(aa.dis.subs, 1:400, scale(as.vector(aa.dis.subs))), Rowv = NA, Colv =
"Rowv", col = color.final, scale = "none", xlab = "reference + type", ylab =
            "alternative + type", main = "Uniprot1 - Disease causing substitutions
(normalized)",
          add.expr = polygon(area, lwd = 3, border = 3), dendrogram = "none", trace = "none",
density.info = "none")
legend("bottomleft",c("neg - negative","pho - hydrophobic","pol - polar","pos - positive"),
cex = 0.65)

# Genetic code 2 or more mutations
res.diss.save <- res.diss
res.nondiss.save <- res.nondiss
sum(diag(codes.aa.res))/sum(codes.aa.res)
observed.multi.count <- sum(res * ifelse(codes.aa.res == 0, 1, 0))
observer.multi.perc <- sum(res * ifelse(codes.aa.res == 0, 1, 0))/sum(res)
diss.multi.count <- sum(res.diss.save * ifelse(codes.aa.res == 0, 1, 0))
diss.multi.perc <- diss.multi.count/sum(res.diss.save)
nondiss.multi.count <- sum(res.nondiss.save * ifelse(codes.aa.res == 0, 1, 0))
nondiss.multi.perc <- nondiss.multi.count/sum(res.nondiss.save)
multi.changes <- c(diss.multi.count, nondiss.multi.count)
names(multi.changes) <- c("dis","nondis")
multi.changes
expected.multi <-
c(sum(res.diss.save),sum(res.nondiss.save))/(sum(res.diss.save)+sum(res.nondiss.save)) *
sum(multi.changes)
names(expected.multi) <- c("dis","nondis")
multi <- data.frame(multi.changes,expected.multi)
chisq.test(multi.changes, p = expected.multi, rescale.p = T)
chisq.test(multi)
colnames(multi) <- c("observed","expected")

tt <- codes.aa.res
rownames(tt) <- type
colnames(tt) <- type
diag(tt) <- 1
k <- arrayInd(which(tt == 0), dim(tt))
multi.type.perc <- sum(rownames(tt)[k[,1]] == colnames(tt)[k[,2]])/nrow(k)
diag(tt) <- 0
k <- arrayInd(which(tt != 0), dim(tt))
```

```
single.type.perc <- sum(rownames(tt)[k[,1]] == colnames(tt)[k[,2]])/nrow(k)

# Genetic code by position
for(j in 1:3)
  assign(paste("one",j, sep = ""), sapply(t0, codeSwitch, position = 1, depth = j))
for(j in 1:3)
  assign(paste("two",j, sep = ""), sapply(t0, codeSwitch, position = 2, depth = j))
for(j in 1:3)
  assign(paste("three",j, sep = ""), sapply(t0, codeSwitch, position = 3, depth = j))
codes.one.aa <- data.frame(g[t0], g[one1], g[one2], g[one3])
codes.two.aa <- data.frame(g[t0], g[two1], g[two2], g[two3])
codes.three.aa <- data.frame(g[t0], g[three1], g[three2], g[three3])
codes.one.aa.res <- table(codes.one.aa[[1]], codes.one.aa[[2]]) + table(codes.one.aa[[1]],
codes.one.aa[[3]]) + table(codes.one.aa[[1]], codes.one.aa[[4]])
codes.two.aa.res <- table(codes.two.aa[[1]], codes.two.aa[[2]]) + table(codes.two.aa[[1]],
codes.two.aa[[3]]) + table(codes.two.aa[[1]], codes.two.aa[[4]])
codes.three.aa.res <- table(codes.three.aa[[1]], codes.three.aa[[2]]) +
table(codes.three.aa[[1]], codes.three.aa[[3]]) + table(codes.three.aa[[1]], codes.three.aa

[[4]])
codes.one.aa.res <- t(codes.one.aa.res[-1,-1])
codes.two.aa.res <- t(codes.two.aa.res[-1,-1])
codes.three.aa.res <- t(codes.three.aa.res[-1,-1])
rownames(codes.one.aa.res) <-
c("Ala","Cys","Asp","Glu","Phe","Gly","His","Ile","lys","Leu","Met","Asn","Pro","Gln","Arg","
Ser","Thr","Val","Trp","Tyr")
rownames(codes.two.aa.res) <-
c("Ala","Cys","Asp","Glu","Phe","Gly","His","Ile","lys","Leu","Met","Asn","Pro","Gln","Arg","
Ser","Thr","Val","Trp","Tyr")
rownames(codes.three.aa.res) <-
c("Ala","Cys","Asp","Glu","Phe","Gly","His","Ile","lys","Leu","Met","Asn","Pro","Gln","Arg","
Ser","Thr","Val","Trp","Tyr")
colnames(codes.one.aa.res) <-
c("Ala","Cys","Asp","Glu","Phe","Gly","His","Ile","lys","Leu","Met","Asn","Pro","Gln","Arg","
Ser","Thr","Val","Trp","Tyr")
colnames(codes.two.aa.res) <-
c("Ala","Cys","Asp","Glu","Phe","Gly","His","Ile","lys","Leu","Met","Asn","Pro","Gln","Arg","
Ser","Thr","Val","Trp","Tyr")
colnames(codes.three.aa.res) <-
c("Ala","Cys","Asp","Glu","Phe","Gly","His","Ile","lys","Leu","Met","Asn","Pro","Gln","Arg","
Ser","Thr","Val","Trp","Tyr")
codes.one.aa.res <- codes.one.aa.res[order(rownames(codes.one.aa.res)),
order(colnames(codes.one.aa.res))]
codes.two.aa.res <- codes.two.aa.res[order(rownames(codes.two.aa.res)),
order(colnames(codes.two.aa.res))]
codes.three.aa.res <- codes.three.aa.res[order(rownames(codes.three.aa.res)),
order(colnames(codes.three.aa.res))]

# expected synonymous by position:
sum(diag(codes.one.aa.res))/sum(codes.one.aa.res)
sum(diag(codes.two.aa.res))/sum(codes.two.aa.res)
sum(diag(codes.three.aa.res))/sum(codes.three.aa.res)

codes.aa.res.save <- codes.aa.res
diag(codes.aa.res.save) <- 0
gcp11 <- sum(ifelse((codes.one.aa.res != 0 & codes.two.aa.res == 0 & codes.three.aa.res ==
0), 1, 0) * codes.one.aa.res)/sum(codes.aa.res.save)
gcp12 <- sum(ifelse((codes.one.aa.res != 0 & codes.two.aa.res == 0 & codes.three.aa.res ==
0), 1, 0) * res)/sum(res)
gcp13 <- sum(ifelse((codes.one.aa.res != 0 & codes.two.aa.res == 0 & codes.three.aa.res ==
0), 1, 0) * res.diss)/sum(res.diss)
gcp14 <- sum(ifelse((codes.one.aa.res != 0 & codes.two.aa.res == 0 & codes.three.aa.res ==
0), 1, 0) * res.nondiss)/sum(res.nondiss)
gcp21 <- sum(ifelse((codes.two.aa.res != 0 & codes.one.aa.res == 0 & codes.three.aa.res ==
0), 1, 0) * codes.two.aa.res)/sum(codes.aa.res.save)
gcp22 <- sum(ifelse((codes.two.aa.res != 0 & codes.one.aa.res == 0 & codes.three.aa.res ==
0), 1, 0) * res)/sum(res)
gcp23 <- sum(ifelse((codes.two.aa.res != 0 & codes.one.aa.res == 0 & codes.three.aa.res ==
0), 1, 0) * res.diss)/sum(res.diss)
gcp24 <- sum(ifelse((codes.two.aa.res != 0 & codes.one.aa.res == 0 & codes.three.aa.res ==
0), 1, 0) * res.nondiss)/sum(res.nondiss)
diag(codes.three.aa.res) <- 0
gcp31 <- sum(ifelse((codes.three.aa.res != 0 & codes.two.aa.res == 0 & codes.one.aa.res ==
0), 1, 0) * codes.three.aa.res)/sum(codes.aa.res.save)
gcp32 <- sum(ifelse((codes.three.aa.res != 0 & codes.two.aa.res == 0 & codes.one.aa.res ==
0), 1, 0) * res)/sum(res)
```

```
gcp33 <- sum(ifelse((codes.three.aa.res != 0 & codes.two.aa.res == 0 & codes.one.aa.res ==
0), 1, 0) * res.diss)/sum(res.diss)
gcp34 <- sum(ifelse((codes.three.aa.res != 0 & codes.two.aa.res == 0 & codes.one.aa.res ==
0), 1, 0) * res.nondiss)/sum(res.nondiss)
gcp <-
data.frame(c(gcp11,gcp12,gcp13,gcp14),c(gcp21,gcp22,gcp23,gcp24),c(gcp31,gcp32,gcp33,gcp34))
rownames(gcp) <- c("expected","observed","dis","nondis")
colnames(gcp) <- c("1st base", "2nd base", "3rd base")
gcp

# Genectic code by position - test
prob.expected <- c(gcp11,gcp21,gcp31)
prob.dis <- c(sum(res.diss),sum(res.nondiss))
observed <- c(sum(ifelse((codes.one.aa.res != 0 & codes.two.aa.res == 0 & codes.three.aa.res
== 0), 1, 0) * res),sum(ifelse((codes.two.aa.res != 0 & codes.one.aa.res

== 0 & codes.three.aa.res == 0), 1, 0) * res),sum(ifelse((codes.three.aa.res != 0 &
codes.two.aa.res == 0 & codes.one.aa.res == 0), 1, 0) * res))
chisq.test(observed, p = prob.expected, rescale.p = T)$p.value
expected <- sum(observed) * (prob.expected/sum(prob.expected))
dis.1 <- sum(ifelse((codes.one.aa.res != 0 & codes.two.aa.res == 0 & codes.three.aa.res ==
0), 1, 0) * res.diss)
nondis.1 <- sum(ifelse((codes.one.aa.res != 0 & codes.two.aa.res == 0 & codes.three.aa.res ==
0), 1, 0) * res.nondiss)
dis.2 <- sum(ifelse((codes.two.aa.res != 0 & codes.one.aa.res == 0 & codes.three.aa.res ==
0), 1, 0) * res.diss)
nondis.2 <- sum(ifelse((codes.two.aa.res != 0 & codes.one.aa.res == 0 & codes.three.aa.res ==
0), 1, 0) * res.nondiss)
dis.3 <- sum(ifelse((codes.three.aa.res != 0 & codes.two.aa.res == 0 & codes.one.aa.res ==
0), 1, 0) * res.diss)
nondis.3 <- sum(ifelse((codes.three.aa.res != 0 & codes.two.aa.res == 0 & codes.one.aa.res ==
0), 1, 0) * res.nondiss)
chisq.test(c(dis.1,nondis.1), p = prob.dis, rescale.p = T)$p.value
chisq.test(c(dis.2,nondis.2), p = prob.dis, rescale.p = T)$p.value
chisq.test(c(dis.3,nondis.3), p = prob.dis, rescale.p = T)$p.value

chisq.test(res[res > 0 & codes.aa.res > 0], p = codes.aa.res[res > 0 & codes.aa.res > 0],
rescale.p = T)$p.value

suppressWarnings(chisq.test(res.diss.save[res.nondiss.save > 0], p =
res.nondiss.save[res.nondiss.save > 0], rescale.p = T))

# pairs
# need to run Uniprot2 befor this
res.diss.combined <- res.diss + t(res.diss)
ress.combinded <- ress + t(ress)
combined <- res.diss.combined/ress.combinded
combined[lower.tri(combined)] <- NaN
comb <- data.frame(expand.grid(rownames(combined), colnames(combined)), as.vector(combined))
comb <- comb[!is.nan(comb[,3]),]
comb[,4] <- 76-rank(comb[,3])
comb2 <- read.table("supplement2")
comb <- as.data.frame(bind_cols(comb, comb2[,3:4]))
comb[,7] <- (comb[,4] + comb[,6])/2
comb <- comb[order(comb[,7]),c(2,1,3,4,5,6,7)]
rownames(comb) <- NULL
colnames(comb) <- c("amino acid","amino acid","disese causing percentage, Uniprot1","rank,
Uniprot1","disease causing percentage, Uniprot2","rank, Uniprot2","average
                    rank")

##############################################################################################
# Uniprot2
##############################################################################################
load <- read.table("uniprot2.txt", sep = "\t", header = F, skip = 144, stringsAsFactors = F,
quote = "", fill = T)
load <- load[grepl(">",load$V10),]
data1 <- load[!duplicated(load$V10),]
data2 <- load[!duplicated(load[c("V10", "V13")]),]

# subset
dis.string <- c("pathogenic","not provided,pathogenic","uncertain
significance,pathogenic","likely pathogenic,pathogenic","pathogenic,other")
data2.dis <- data2[data2$V6 %in% dis.string,]
data1.dis <- data1[data1$V6 %in% dis.string,]
data2.nondis <- data2[!(data2$V6 %in% dis.string),]
data1.nondis <- data1[!(data1$V6 %in% dis.string),]
```

```
# res
old <- substr(load[,3], 3, 5)
new <- sub(".{5}[[:digit:]]*", "", load[,3])
new <- substr(new, 1, 3)
res.full <- t(table(old,new)[-5,])
res <- res.full[-17,-17]

old <- substr(data1[,3], 3, 5)
new <- sub(".{5}[[:digit:]]*", "", data1[,3])
new <- substr(new, 1, 3)
res1.full <- t(table(old,new)[-5,])
res1 <- res1.full[-17,-17]

old <- substr(data2[,3], 3, 5)
new <- sub(".{5}[[:digit:]]*", "", data2[,3])
new <- substr(new, 1, 3)
res2.full <- t(table(old,new)[-5,])
res2 <- res2.full[-17,-17]

chisq.test(res1[res2 > 0], p = res2[res2 > 0], rescale.p = T)$p.value

# heatmap
res.x <- res1.full
res.x[res.x < 10] <- NA
heatmap.2(res.x, Rowv = F, Colv = "Rowv", col = color.final, dendrogram = "none", trace =
"none", density.info = "none", xlab = "reference", ylab = "alternative",
          key.xlab = "count", main = "Uniprot2 map")

# normalization
type <-
c("pho","pos","pol","neg","pol","pol","neg","pho","pos","pho","pho","pos","pho","pho","pho","
pol","pol","pho","pol","pho")
res.type2 <- rowsum(t(rowsum(t(res), type)), type)
codes.aa.res.type <- codes.aa.res
diag(codes.aa.res.type) <- 0
codes.aa.res.type[codes.aa.res.type > 0] <- 1
codes.aa.res.type <- rowsum(t(rowsum(t(codes.aa.res.type), type)), type)
codes.aa.res.type.full <- rowsum(t(rowsum(t(matrix(1, 20, 20)), type)), type)
heatmap.2(res.type2/codes.aa.res.type, Rowv = F, Colv = "Rowv", col = color.final, dendrogram
= "none", trace = "none", density.info = "none", xlab = "reference type",
          ylab = "alternative type", main = "Structural classification count map - Uniprot2")
legend("bottomleft",c("neg - negative","pho - hydrophobic","pol - polar","pos -
positive"),cex=0.7)

# res subset
old <- substr(data1.dis[,3], 3, 5)
new <- sub(".{5}[[:digit:]]*", "", data1.dis[,3])
new <- substr(new, 1, 3)
res1.dis.full <- t(table(old,new)[-5,])
res1.dis <- res1.dis.full[-17,-17]
old <- substr(data1.nondis[,3], 3, 5)
new <- sub(".{5}[[:digit:]]*", "", data1.nondis[,3])
new <- substr(new, 1, 3)
res1.nondis.full <- t(table(old,new)[-5,])
res1.nondis <- res1.nondis.full[-17,-17]
old <- substr(data2.dis[,3], 3, 5)
new <- sub(".{5}[[:digit:]]*", "", data2.dis[,3])
new <- substr(new, 1, 3)
res2.dis.full <- t(table(old,new)[-5,])
res2.dis <- res2.dis.full[-17,-17]
old <- substr(data2.nondis[,3], 3, 5)
new <- sub(".{5}[[:digit:]]*", "", data2.nondis[,3])
new <- substr(new, 1, 3)
res2.nondis.full <- t(table(old,new)[-5,])
res2.nondis <- res2.nondis.full[-17,-17]

# Type
res.type.diss.1 <- rowsum(t(rowsum(t(res1.dis), type)), type)
res.type.diss.2 <- rowsum(t(rowsum(t(res2.dis), type)), type)
res.type.diss.1 <- res.type.diss.1/res.type2
res.type.diss.2 <- res.type.diss.2/res.type2
heatmap.2(res.type.diss.1, Rowv = F, Colv = "Rowv", col = color.final, dendrogram = "none",
trace = "none", density.info = "none", xlab = "reference type", ylab =
            "alternative type", key.xlab = "disease causing percantage",main = "Structural
classification disease map - Uniprot2")
legend("bottomleft",c("neg - negative","pho - hydrophobic","pol - polar","pos - positive"),
cex = 0.7)
```

```
# dis AA
barplot(colSums(res1.dis.full)/colSums(res1.full), xlab = "reference amino acid", ylab =
"pecentage of disease causing substitutions", ylim = c(0,0.025), main =
          "Uniprot2")
barplot(rowSums(res1.dis.full)/rowSums(res1.full), xlab = "alternative amino acid", ylab =
"pecentage of disease causing substitutions", ylim = c(0,0.05), main =
          "Uniprot2")
barplot(rowSums(res1.dis)/rowSums(res1), xlab = "alternative amino acid", ylab = "pecentage
of disease causing substitutions", ylim = c(0,0.01), main = "Uniprot2")

# dis heatmap
ress <- res1
aa.dis.subs <- res1.dis/ress
rownames(aa.dis.subs) <- paste(type, rownames(aa.dis.subs))
colnames(aa.dis.subs) <- paste(type, colnames(aa.dis.subs))
aa.dis.subs <- aa.dis.subs[order(rownames(aa.dis.subs)), order(colnames(aa.dis.subs))]
aa.dis.subs
heatmap.2(aa.dis.subs, Rowv = NA, Colv = "Rowv", col = color.final, scale = "none", xlab =
"reference + type", ylab = "alternative + type", main = "Uniprot2 - Disease
          causing substitutions",
          add.expr = polygon(area, lwd = 3, border = 3), dendrogram = "none", trace = "none",
density.info = "none", key.xlab = "percentage of disease causing")
legend("bottomleft",c("neg - negative","pho - hydrophobic","pol - polar","pos - positive"),
cex = 0.65)
# normalized:
heatmap.2(replace(aa.dis.subs, 1:400, scale(as.vector(aa.dis.subs))), Rowv = NA, Colv =
"Rowv", col = color.final, scale = "none", xlab = "reference + type", ylab =
            "alternative + type", main = "Uniprot2 - Disease causing substitutions
(normalized)",
          add.expr = polygon(area, lwd = 3, border = 3), dendrogram = "none", trace = "none",
density.info = "none")
legend("bottomleft",c("neg - negative","pho - hydrophobic","pol - polar","pos - positive"),
cex = 0.65)

# Genetic code chi sq test
chisq.test(res1[res1 > 0], p = codes.aa.res[res1 > 0], rescale.p = T)
chisq.test(res2, p = codes.aa.res, rescale.p = T)

# Genectic code by position - test
prob.expected <- c(gcp11,gcp21,gcp31)
prob.dis <- c(sum(res1.dis),sum(res1.nondis))
observed <- c(sum(ifelse((codes.one.aa.res != 0 & codes.two.aa.res == 0 & codes.three.aa.res
== 0), 1, 0) * res1),sum(ifelse((codes.two.aa.res != 0 & codes.one.aa.res

== 0 & codes.three.aa.res == 0), 1, 0) * res1),sum(ifelse((codes.three.aa.res != 0 &
codes.two.aa.res == 0 & codes.one.aa.res == 0), 1, 0) * res1))
chisq.test(observed, p = prob.expected, rescale.p = T)$p.value
expected <- sum(observed) * (prob.expected/sum(prob.expected))
dis.1 <- sum(ifelse((codes.one.aa.res != 0 & codes.two.aa.res == 0 & codes.three.aa.res ==
0), 1, 0) * res1.dis)
nondis.1 <- sum(ifelse((codes.one.aa.res != 0 & codes.two.aa.res == 0 & codes.three.aa.res ==
0), 1, 0) * res1.nondis)
dis.2 <- sum(ifelse((codes.two.aa.res != 0 & codes.one.aa.res == 0 & codes.three.aa.res ==
0), 1, 0) * res1.dis)
nondis.2 <- sum(ifelse((codes.two.aa.res != 0 & codes.one.aa.res == 0 & codes.three.aa.res ==
0), 1, 0) * res1.nondis)
dis.3 <- sum(ifelse((codes.three.aa.res != 0 & codes.two.aa.res == 0 & codes.one.aa.res ==
0), 1, 0) * res1.dis)
nondis.3 <- sum(ifelse((codes.three.aa.res != 0 & codes.two.aa.res == 0 & codes.one.aa.res ==
0), 1, 0) * res1.nondis)
chisq.test(c(dis.1,nondis.1), p = prob.dis, rescale.p = T)$p.value
chisq.test(c(dis.2,nondis.2), p = prob.dis, rescale.p = T)$p.value
chisq.test(c(dis.3,nondis.3), p = prob.dis, rescale.p = T)$p.value
gcpp <- t(data.frame(as.character(prob.expected), as.character(observed),
as.character(c(dis.1,dis.2,dis.3)),as.character(c(nondis.1,nondis.2,nondis.3))))
prob.expected/sum(prob.expected) * sum(observed)
prob.dis/sum(prob.dis) * sum(c(dis.3+nondis.3))

# res1.dis correlation test
sim.res.save <- res1
res1 <- res1.dis/res1
real.cor <- cor(res1[upper.tri(res1) & !is.na(res1)], t(res1)[upper.tri(res1) &
!is.na(res1)])
sim.cor <- function(){
  ss <- sample(res1[!is.na(res1)],150)
  return(cor(ss[1:75],ss[76:150]))
```

```
}
set.seed(1)
test.cor <- replicate(1000000, sim.cor())
plot(density(c(real.cor,test.cor)), xlab = "Correlation", main = "Correlation simulation")
points(real.cor,0, col = 2)
legend("topright","real correlation",col=2,pch=1)
(test.cor.p <- (sum(real.cor < test.cor)+1)/(1000001))
res1 <- sim.res.save
cor.test(res1[upper.tri(res1) & !is.na(res1)], t(res1)[upper.tri(res1) & !is.na(res1)],
alternative = "g")$p.value

# combined
res.diss.combined <- res1.dis + t(res1.dis)
ress.combinded <- res1 + t(res1)
combined <- res.diss.combined/ress.combinded
combined[lower.tri(combined)] <- NaN
comb <- data.frame(expand.grid(rownames(combined), colnames(combined)), as.vector(combined))
comb <- comb[!is.nan(comb[,3]),]
comb[,4] <- 76-rank(comb[,3])
write.table(comb, "supplement2")
comb <- comb[order(comb[,4]),c(2,1,3,4)]

##############################################################################################
# 1000G
##############################################################################################
# !!setwd to location of 1000g dataset
# !!extract files in the 1000g folder
res <- as.data.frame(matrix(ncol = 7, nrow = 0))
colnames(res) <- c("old","new","rs","pp","pp.sc","sift","sift.sc")
res[,1:7] <- sapply(res[,1:7], as.character)
sum.miss <- 0
sum.syn <- 0

for(i in 1:22){
  file.name <-
paste("ALL.chr",i,".phase3_shapeit2_mvncall_integrated_v5.20130502.sites.annotation.vcf", sep
= "")
  load <- read.table(file.name, header = F, skip = 258, sep = "", stringsAsFactors = F)
  t <- (grepl("missense", load[,8]) & nchar(load$V4) == 1 & nchar(load$V5) == 1)
  tt <- (grepl("synonymous", load[,8]) & nchar(load$V4) == 1 & nchar(load$V5) == 1)
  sum.miss <- sum.miss + sum(t)
  sum.syn <- sum.syn + sum(tt)
  data <- subset(load, subset = t)
  rs <- data[,3]
  pp <- as.character(lapply(strsplit(data[,8], split = "\\|"), "[", 18))
  pp.sc <- sapply(strsplit(pp, split = "[\\(|\\)]"), "[", 2)
  pp <- sapply(strsplit(pp, split = "[\\(|\\)]"), "[", 1)
  sift <- as.character(lapply(strsplit(data[,8], split = "\\|"), "[", 17))
  sift.sc <- sapply(strsplit(sift, split = "[\\(|\\)]"), "[", 2)
  sift <- sapply(strsplit(sift, split = "[\\(|\\)]"), "[", 1)
  data <- strsplit(data[,8], split = "\\/")
  old <- as.character(lapply(data, "[[", 2))
  new <- as.character(lapply(data, "[[", 1))
  old <- substr(old, 1, 1)
  new <- substr(new, nchar(new), nchar(new))
  res <- bind_rows(res, data.frame(old, new, rs, pp, pp.sc, sift, sift.sc))
}

colnames(res) <- c("new", "old", "rs", "pp","pp.sc","sift","sift.sc")
res.full <- table(res[,1:2])[,-18]
colnames(res.full) <-
c("Ala","Cys","Asp","Glu","Phe","Gly","His","Ile","lys","Leu","Met","Asn","Pro","Gln","Arg","
Ser","Thr","Val","Trp","Tyr")
rownames(res.full) <-
c("Ala","Cys","Asp","Glu","Phe","Gly","His","Ile","lys","Leu","Met","Asn","Pro","Gln","Arg","
Ser","Thr","Val","Trp","Tyr")
res.full <- res.full[order(rownames(res.full)), order(colnames(res.full))]
res.save <- res

# synonymous
sum(diag(codes.aa.res))/sum(codes.aa.res)
sum.syn/(sum.syn + sum.miss)

# chisq test
diag(codes.aa.res) <- 0
chisq.test(x = res.full[codes.aa.res != 0], p = codes.aa.res[codes.aa.res != 0], rescale.p =
T)
```

```
# scores
t <- scan("supplement3.txt", what = "character", sep = "")
res.scores <- res.save[res.save$rs %in% t,]
res.dis <- table(res.scores$new, res.scores$old)
rownames(res.dis) <-
c("Ala","Cys","Asp","Glu","Phe","Gly","His","Ile","lys","Leu","Met","Asn","Pro","Gln","Arg","
Ser","Thr","Val","Trp","Tyr")
colnames(res.dis) <-
c("Ala","Cys","Asp","Glu","Phe","Gly","His","Ile","lys","Leu","Met","Asn","Pro","Gln","Arg","
Ser","Thr","Val","Trp","Tyr")
res.dis <- res.dis[order(rownames(res.dis)), order(colnames(res.dis))]

par(mfrow=c(2,2))
plot(density(as.numeric(res.save$pp.sc), na.rm = T), col = 11, xlab = "PolyPhen value", main
= "PolyPhen prediction - Scores")
lines(density(as.numeric(res.scores$pp.sc), na.rm = T), col = 2)
legend(x = "topright", legend = c("all","disease"), col = c(11, 2), lty = 1, cex = 0.7)
barplot(rbind((table(res.scores$pp)/sum(table(res.scores$pp))),(table(res.save$pp)/sum(table(
res.save$pp)))), col = c(2,3), beside = T, xlab = "PolyPhen class",
        main = "PolyPhen prediction - Classes", ylab = "Percentage of data classified",
cex.names = 0.6)
legend(x = "topright", legend = c("all","disease"), col = c(3, 2), pch = 16, cex = 0.7)
# par(mfrow=c(1,2))
plot(density(as.numeric(res.save$sift.sc), na.rm = T), col = 11, xlab = "SIFT value", main =
"SIFT prediction - Scores")
lines(density(as.numeric(res.scores$sift.sc), na.rm = T), col = 2)
legend(x = "topright", legend = c("all","disease"), col = c(11, 2), lty = 1, cex = 0.7)
barplot(rbind((table(res.scores$sift)/sum(table(res.scores$sift))),(table(res.save$sift)/sum(
table(res.save$sift)))), col = c(2,3), beside = T, xlab = "SIFT class",
        main = "SIFT prediction - Classes", ylab = "Percentage of data classified")
legend(x = "top", legend = c("all","disease"), col = c(3, 2), pch = 16, cex = 0.7)
par(mfrow=c(1,1))

# scores testing
wilcox.test(x = as.numeric(res.scores$pp.sc), y = as.numeric(res.save$pp.sc), alternative =
"greater", paired = F, conf.int = T)$p.value
wilcox.test(x = as.numeric(res.scores$sift.sc), y = as.numeric(res.save$sift.sc), alternative
= "l", paired = F, conf.int = T)$p.value
chisq.test(table(res.scores$sift), p = table(res.save$sift), rescale.p = T)$p.value
test.pp.save <- c(table(res.save$pp)[1], sum(table(res.save$pp)[3]))
test.pp.scores <- c(table(res.scores$pp)[1], sum(table(res.scores$pp)[3]))
chisq.test(test.pp.scores, p = test.pp.save, rescale.p = T)$p.value

# population
res <- as.data.frame(matrix(ncol = 7, nrow = 0))
colnames(res) <- c("old","new","eas","amr","afr","eur","sas")
res[,1:7] <- sapply(res[,1:7], as.character)
for(i in 1:22){
  file.name <-
paste("ALL.chr",i,".phase3_shapeit2_mvncall_integrated_v5.20130502.sites.annotation.vcf", sep
= "")
  load <- read.table(file.name, header = F, skip = 258, sep = "", stringsAsFactors = F)
  t <- (grepl("missense", load[,8]) & nchar(load$V4) == 1 & nchar(load$V5) == 1)
  data <- subset(load, subset = t)
  data2 <- strsplit(data[,8], split = "\\/")
  new <- as.character(lapply(data2, "[[", 2))
  old <- as.character(lapply(data2, "[[", 1))
  new <- substr(new, 1, 1)
  old <- substr(old, nchar(old), nchar(old))
  eas <- sapply(strsplit(data[,8], split = "EAS_AF="),"[",2)
  eas <- sapply(strsplit(eas, split = "\\;"),"[",1)
  amr <- sapply(strsplit(data[,8], split = "AMR_AF="),"[",2)
  amr <- sapply(strsplit(amr, split = "\\;"),"[",1)
  afr <- sapply(strsplit(data[,8], split = "AFR_AF="),"[",2)
  afr <- sapply(strsplit(afr, split = "\\;"),"[",1)
  eur <- sapply(strsplit(data[,8], split = "EUR_AF="),"[",2)
  eur <- sapply(strsplit(eur, split = "\\;"),"[",1)
  sas <- sapply(strsplit(data[,8], split = "SAS_AF="),"[",2)
  sas <- sapply(strsplit(sas, split = "\\;"),"[",1)
  res <- bind_rows(res, data.frame(old, new, eas, amr, afr, eur, sas))
}

zz1 <- as.matrix(res[,3:7])
mode(zz1) <- "numeric"
sum(rowMeans(zz1))
```

```
res <- res[-which(res$eas > 0.5 | res$amr > 0.5 | res$afr > 0.5 | res$eur > 0.5 | res$sas >
0.5),]
res <- res[-which(res$old == "U" | res$new == "U"),]

zz2 <- as.matrix(res[,3:7])
mode(zz2) <- "numeric"
sum(rowMeans(zz2))

x.eas <- group_by(res, old, new) %>% summarize(mean(as.numeric(eas)))
x.afr <- group_by(res, old, new) %>% summarize(mean(as.numeric(afr)))
x.amr <- group_by(res, old, new) %>% summarize(mean(as.numeric(amr)))
x.eur <- group_by(res, old, new) %>% summarize(mean(as.numeric(eur)))
x.sas <- group_by(res, old, new) %>% summarize(mean(as.numeric(sas)))

res.all <- table(res$new, res$old)
x.res.eas <- res.all
for(i in 1:length(x.eas$old))
  x.res.eas[x.eas$new[i],x.eas$old[i]] <- as.numeric(x.eas[x.eas$old == x.eas$old[i] &
x.eas$new == x.eas$new[i],][3])
x.res.amr <- res.all
for(i in 1:length(x.amr$old))
  x.res.amr[x.amr$new[i],x.amr$old[i]] <- as.numeric(x.amr[x.amr$old == x.amr$old[i] &
x.amr$new == x.amr$new[i],][3])
x.res.afr <- res.all
for(i in 1:length(x.afr$old))
  x.res.afr[x.afr$new[i],x.afr$old[i]] <- as.numeric(x.afr[x.afr$old == x.afr$old[i] &
x.afr$new == x.afr$new[i],][3])
x.res.eur <- res.all
for(i in 1:length(x.eur$old))
  x.res.eur[x.eur$new[i],x.eur$old[i]] <- as.numeric(x.eur[x.eur$old == x.eur$old[i] &
x.eur$new == x.eur$new[i],][3])
x.res.sas <- res.all
for(i in 1:length(x.sas$old))
  x.res.sas[x.sas$new[i],x.sas$old[i]] <- as.numeric(x.sas[x.sas$old == x.sas$old[i] &
x.sas$new == x.sas$new[i],][3])
xx.res.all <- x.res.eas + x.res.amr + x.res.afr + x.res.eur + x.res.sas

col.break <- c(seq(0,0.2,length=100),seq(0.201,0.25,length=100),seq(0.251,0.5,length=100))
xx <- x.res.eas/xx.res.all
rownames(xx) <-
c("Ala","Cys","Asp","Glu","Phe","Gly","His","Ile","lys","Leu","Met","Asn","Pro","Gln","Arg","
Ser","Thr","Val","Trp","Tyr")
colnames(xx) <-
c("Ala","Cys","Asp","Glu","Phe","Gly","His","Ile","lys","Leu","Met","Asn","Pro","Gln","Arg","
Ser","Thr","Val","Trp","Tyr")
heatmap.2(xx, Rowv = NA, Colv = "Rowv", col = color.final, scale = "none", xlab =
"reference", ylab = "alternative", main = "eas", dendrogram = "none", trace = "none",
          density.info = "none", breaks = col.break, key.xlab = "percentage in eas")
xx <- x.res.amr/xx.res.all
rownames(xx) <-
c("Ala","Cys","Asp","Glu","Phe","Gly","His","Ile","lys","Leu","Met","Asn","Pro","Gln","Arg","
Ser","Thr","Val","Trp","Tyr")
colnames(xx) <-
c("Ala","Cys","Asp","Glu","Phe","Gly","His","Ile","lys","Leu","Met","Asn","Pro","Gln","Arg","
Ser","Thr","Val","Trp","Tyr")
heatmap.2(xx, Rowv = NA, Colv = "Rowv", col = color.final, scale = "none", xlab =
"reference", ylab = "alternative", main = "amr", dendrogram = "none", trace = "none",
          density.info = "none", breaks = col.break, key.xlab = "percentage in amr")
xx <- x.res.afr/xx.res.all
rownames(xx) <-
c("Ala","Cys","Asp","Glu","Phe","Gly","His","Ile","lys","Leu","Met","Asn","Pro","Gln","Arg","
Ser","Thr","Val","Trp","Tyr")
colnames(xx) <-
c("Ala","Cys","Asp","Glu","Phe","Gly","His","Ile","lys","Leu","Met","Asn","Pro","Gln","Arg","
Ser","Thr","Val","Trp","Tyr")
heatmap.2(xx, Rowv = NA, Colv = "Rowv", col = color.final, scale = "none", xlab =
"reference", ylab = "alternative", main = "afr", dendrogram = "none", trace = "none",
          density.info = "none", breaks = col.break, key.xlab = "percentage in afr")
xx <- x.res.eur/xx.res.all
rownames(xx) <-
c("Ala","Cys","Asp","Glu","Phe","Gly","His","Ile","lys","Leu","Met","Asn","Pro","Gln","Arg","
Ser","Thr","Val","Trp","Tyr")
colnames(xx) <-
c("Ala","Cys","Asp","Glu","Phe","Gly","His","Ile","lys","Leu","Met","Asn","Pro","Gln","Arg","
Ser","Thr","Val","Trp","Tyr")
heatmap.2(xx, Rowv = NA, Colv = "Rowv", col = color.final, scale = "none", xlab =
"reference", ylab = "alternative", main = "eur", dendrogram = "none", trace = "none",
```

```
          density.info = "none", breaks = col.break, key.xlab = "percentage in eur")
xx <- x.res.sas/xx.res.all
rownames(xx) <-
c("Ala","Cys","Asp","Glu","Phe","Gly","His","Ile","lys","Leu","Met","Asn","Pro","Gln","Arg","
Ser","Thr","Val","Trp","Tyr")
colnames(xx) <-
c("Ala","Cys","Asp","Glu","Phe","Gly","His","Ile","lys","Leu","Met","Asn","Pro","Gln","Arg","
Ser","Thr","Val","Trp","Tyr")
heatmap.2(xx, Rowv = NA, Colv = "Rowv", col = color.final, scale = "none", xlab =
"reference", ylab = "alternative", main = "sas", dendrogram = "none", trace = "none",
          density.info = "none", breaks = col.break, key.xlab = "percentage in sas")

# Difference within population
par(mfrow=c(2,3))
z <- x.res.eas/xx.res.all
plot(density(z[!is.na(z)]),xlim = c(0.1,0.5), xlab = "Percantage of variation", main = "eas")
z <- x.res.amr/xx.res.all
plot(density(z[!is.na(z)]),xlim = c(0.1,0.5), xlab = "Percantage of variation", main = "amr")
z <- x.res.afr/xx.res.all
plot(density(z[!is.na(z)]),xlim = c(0.1,0.5), xlab = "Percantage of variation", main = "afr")
z <- x.res.eur/xx.res.all
plot(density(z[!is.na(z)]),xlim = c(0.1,0.5), xlab = "Percantage of variation", main = "eur")
z <- x.res.sas/xx.res.all
plot(density(z[!is.na(z)]),xlim = c(0.1,0.5), xlab = "Percantage of variation", main = "sas")
```

9.2. List of symbols and abbreviations

Amino acid symbols			
Ala	Alanine	**Leu**	Leucine
Arg	Arginine	**Lys**	Lysine
Asn	Asparagine	**Met**	Methionine
Asp	Aspartic acid (Aspartate)	**Phe**	Phenylalanine
Cys	Cysteine	**Pro**	Proline
Gln	Glutamine	**Ser**	Serine
Glu	Glutamic acid (Glutamate)	**Thr**	Threonine
Gly	Glycine	**Trp**	Tryptophan
His	Histidine	**Tyr**	Tyrosine
Ile	Isoleucine	**Val**	Valine

Other abbreviations			
3C	Chromosome Conformation Capture	**GIGO**	'garbage in, garbage out'
A	Adenine	**GWAS**	Genome Wide Association Studies
AFR	African	**HGP**	Human Genome Project
AMP	Adenosine monophosphate	**MD**	Molecular Dynamics
AMR	American	**NGS**	Next-generation Sequencing
APS	Adenosine-5-phosphosulfate	**nsSNP**	nonsynonymous SNP
ATP	Adenosine triphosphate	**PDB**	Protein Data Bank
bp	base pair	**PDF**	Probability Density Function
C	Cytosine	**PMF**	Probability Mass Function
CDF	Cumulative Distribution Function	**PPi**	Pyrophosphate, $P_2O_7^{4-}$
CNV	Copy-Number Variant	**SAS**	South Asian
cryo-EM	cryo-Electron Microscopy	**SMRT**	Single-Molecule Real Time
dNTP	Nucleoside triphosphate	**SNP**	Single-Nucleotide Polymorphism
EAS	East Asian	**SV**	Structural Variant
emPCR	emulsion PCR	**T**	Thymine
EUR	European	**U**	Uracile
G	Guanine	**ZMW**	Zero-Mode Waveguide

§ 10. Biography

Personal information

Name: Kristijan Vukovic

Date of birth: 25.10.1992.

E-mail: kvukovic6@gmail.com

Education and training

2007. – 2011. XV. Gimnazija (MIOC), Zagreb, Croatia

2011. – 2014. univ. bacc. chem.
Department of Chemistry, Faculty of Science, University of Zagreb, Croatia

2014. – 2016. mag. chem.
Department of Chemistry, Faculty of Science, University of Zagreb, Croatia

2015. – 2016. Computational biology module
Department of Molecular Biology, Faculty of Science, University of Zagreb, Croatia

2013. BEST Course - Biomass for Bioenergy
Transilvania University of Brasov, Romania

2016. Visiting researcher
University of Melbourne, Australia

Honours and awards

2007. – 2011. Participation at the National Competitions in Chemistry and in Mathematics for high school students

2014. Award from Faculty of Science, University of Zagreb, for excellence in undergraduate study

2013. Special Dean's Award for participation in the Department of Chemistry open doors day

2011. – 2016. Recipient of the City of Zagreb scholarship

2016. Endeavour research fellow

Activities in education and popularization of science

2014. – Training high school students for chemistry competitions

2012. – 2015. Participation in the Department of Chemistry open doors days.

Spoken languages

Croatian Mother tongue

English C2

German A2