UNIUNEA EUROPEANĂ

Instrumente Structurale 2014-2020

Programul Operațional Competitivitate

CeCBiD-EOSC

Centru Cloud și Big Data pentru participarea la Cloud-ul European pentru Știință Deschisă (CeCBiD-EOSC)

Raport Științific și Tehnic 2.5

Servicii și aplicații informatice pentru suportul activității de analiză a datelor de secvențiere de nouă generație

Editor: Mihnea Dulea
Autori: George Necula
Dragoș Nicolae Ciobanu-Zabet
Ionut Traian Vasile

Versiunea: Finală (1.0)
Data: 31.07.2023
Distributie: Internă
Cod document: CBD_RST-2.5

Rezumat: Proiectul POC CeCBiD-EOSC a fost implementat de către echipa Departamentului Fizică Computațională și Tehnologia Informației (DFCTI) din IFIN-HH în perioada 2020-2023. Acest document raportează rezultatele obținute în cadrul Subactivității 2.5 – "*Realizarea de servicii si aplicatii informatice pentru suportul activitatii de analiza a datelor de secventiere de noua generatie*". Sunt prezentate fluxurile de lucru dezvoltate in cadrul DFCTI pentru analiza secventelor genomice umane obtinute prin metoda secventierii de noua generatie (NGS). Fluxurile programate sunt testate, optimizate si validate riguros in vederea integrarii lor in serviciile de analiza NGS care sunt puse la dispoziția utilizatorilor in Centrul de Resurse Cloud și Big Data. Pentru furnizarea serviciilor de analiza NGS se programeaza o platforma software unica si o interfata web de acces al utilizatorilor la aceste servicii.

Prefață

Investițiile în infrastructura de Cloud computing și de Big Data în vederea maximizării potențialului de creștere a economiei digitale europene reprezintă una dintre direcțiile prioritare ale Strategiei Pieței Unice Digitale, care a fost stabilită în 2015 de către Comisia Europeană.[1]

Recunoscând capabilitațile de exploatare a fenomenului Big Data oferite de tehnologia Cloud, Comisia a lansat în aprilie 2016 Inițiativa Europeană Cloud[2], a carei implementare se bazeaza pe Cloud-ul European pentru Stiinta Deschisa (*European Open Science Cloud* – EOSC) si pe Infrastructura Europeana de Date (*European Data Infrastructure* – EDI).

In viziunea Comisiei Europene, EOSC trebuie să asigure pentru comunitatea științifică un mediu virtual sigur, deschis, capabil să ofere servicii de stocare, management, analiză, precum și de refolosire a datelor dincolo de frontiere și discipline stiintifice. În acest cadru, s-a recomandat infrastructurilor europene de cercetare (și în primul rând infrastructurilor ESFRI) sa promoveze reutilizarea datelor proprii pentru inovare și în scopuri educaționale prin sprijinirea conectării lor la EOSC.[3]

Parcursul de Implementare a EOSC[4] a stabilit liniile de acțiune pentru crearea unei federații pan-europene a infrastructurilor de date pentru cercetare, care să înlocuiască fragmentarea existentă cu soluții eficiente și ușor de utilizat pentru stocarea, găsirea, partajarea și refolosirea datelor. Direcțiile de acțiune propuse pentru implementarea modelului federalizat al EOSC privesc arhitectura sistemului, administrarea datelor, serviciile, accesul și interfețele de acces, regulile de participare, precum și guvernanța. Arhitectura EOSC cuprinde un nucleu federativ, care include resursele partajate ale EOSC, precum și multiple infrastructuri de date federate angajate în furnizarea de servicii catre EOSC.

Începand din anul 2018, IFIN-HH a contribuit la implementarea infrastructurii EOSC prin intermediul Departamentului Fizică Computațională și Tehnologia Informației (DFCTI, https://cc.ifin.ro), care a participat, în calitate de asociat al coordonatorului, Fundația EGI[5], la proiectul H2020 EOSC-Hub[6] (2018-2020), destinat dezvoltării resurselor și serviciilor Cloud inițiale pentru EOSC. Sarcina DFCTI a fost de a furniza, prin intermediul centrului Cloud CLOUDIFIN[7], resurse pentru susținerea diferitelor comunități de utilizatori, precum și de a contribui cu servicii naționale la catalogul de servicii al EOSC, în conformitate cu regulile de angajare ale proiectului. În acest scop, EOSC-Hub a finanțat activitatea de management și operare a site-ului CLOUDIFIN, asigurând continuitatea furnizării acestor servicii.

Pentru a putea finanța realizarea masei critice de resurse necesară participării la EOSC, DFCTI a propus proiectul CeCBiD-EOSC in cadrul apelului POC 398/2018. Totodată, pentru continuarea implementării serviciilor specifice EOSC, DFCTI participă, incepînd din 2020, la proiectul H2020 EGI-ACE – *„Advanced Computing for EOSC*" (2020-2023), a cărui misiune este de a asigura servicii EOSC gratuite pentru cercetătorii din toate disciplinele științifice care necesită calcule intensive și Big Data. Astfel, proiectele EGI-ACE și CeCBiD-EOSC acționeaza complementar, la nivelul EU și, respectiv, național, pentru realizarea strategiei de integrare a infrastructurii de calcul și de date a IFIN-HH în EOSC.

Obiectivul general al proiectului CeCBiD-EOSC este creșterea capacității de cercetare în scopul ridicării nivelului de competitivitate științifică pe plan internațional al IFIN-HH, prin modernizarea

[1] "*A Digital Single Market Strategy for Europe*" - COM(2015) 192
[2] "*European Cloud Initiative – Building a competitive data and knowledge economy in Europe*" – COM (2016) 178
[3] "*Long-term sustainability of Research Infrastructures*" – SWD(2017) 323
[4] "*Implementation Roadmap for the European Open Science Cloud*" - SWD(2018) 83
[5] Fundatia EGI, https://www.egi.eu/about/egi-foundation/
[6] "*Integrating and managing services for the EOSC"*, https://www.eosc-hub.eu/
[7] Centrul de resurse Cloud al DFCTI, CLOUDIFIN, http://cloudifin.ifin.ro/

infrastructurii Cloud, extinderea infrastructurii masive de date și realizarea unui centru de date cu performanțe înalte, care să fie integrat în Cloud-ul European pentru Știința Deschisă.[8]

Totodata, proiectul propune o soluție tehnică pentru interconectarea la nivel național, în cadrul unui Cloud federalizat, a centrelor de tip Cloud privat dezvoltate în instituții aparținând sistemului de CDI, capabilă să ofere utilizatorilor acces printr-o interfață unică la resurse și servicii furnizate de aceste centre. Implementarea acestei soluții va eficientiza utilizarea resurselor Cloud de către grupurile de cercetători și va stimula semnificativ cooperarea între specialiștii în tehnologii informatice avansate.

Infrastructura realizată în cadrul proiectului va susține dezvoltarea unor activități de CDI în domeniile sistemelor de calcul paralel și distribuit, învățării automatizate, calculului științific și bioinformaticii, cu aplicații relevante pentru fizica materiei condensate, studiul interacției laser-materie, nanofizică și nanoelectronică.

Obiectivele specifice ale proiectului CeCBiD-EOSC au fost următoarele:

1. Realizarea unui centru performant de resurse Cloud și Big Data prin achiziționarea și instalarea de active corporale și necorporale necesare pentru derularea activităților de CDI prevăzute în proiect.
2. Dezvoltarea și diversificarea serviciilor furnizate la nivel european de catre centrul CLOUDIFIN în perspectiva integrării acestuia în EOSC.
3. Realizarea în cadrul centrului de resurse Cloud a unei soluții tehnice capabilă să interconecteze la nivel național, în cadrul unui sistem federalizat, centrele de tip Cloud privat dezvoltate în instituții aparținând sistemului de CDI, capabilă să ofere utilizatorilor acces printr-o interfață unica la resurse și servicii furnizate de către aceste centre.
4. Asigurarea condițiilor de susținere informațională în tehnologie Cloud și Big Data a participării institutului la colaborări internaționale de anvergură, precum și a noilor opțiuni strategice privind angajarea în direcții de cercetare emergente din spațiul științific internațional, cu relevanță socio-economică deosebită.
5. Dezvoltarea și implementarea de servicii și aplicații informatice pentru administrarea și funcționarea centrului de resurse, care utilizeaza tehnologii Cloud și Big Data pentru: satisfacerea cerințelor IT din faza operațională a proiectului Extreme Light Infrastructure - Nuclear Physics (ELI-NP); modelarea și simularea la nivel molecular a nano- și biosistemelor complexe; analiza datelor de secvențiere de nouă generație.
6. Realizarea condițiilor tehnice și asigurarea suportului de specialitate pentru obținerea și/sau îmbunătățirea de către parteneri economici, în special din cadrul Clusterul Tehnologic Magurele (MHTC), a unor produse și servicii în domenii de specializare inteligentă, care vor conduce la creșterea competitivității acestora.
7. Formarea și perfecționarea personalului calificat, precum și transferul de cunoștințe în domeniile Cloud computing și Big Data către personalul științific și tehnic din alte entitati ale sistemului de CDI.
8. Diseminarea rezultatelor proiectului prin participarea la conferințe (inter)naționale și publicarea de articole științifice în parteneriat public-privat.

Prin obiectivele sale, proiectul conduce la extinderea capacității resurselor Cloud și Big Data, precum și la îmbunătățirea calitativă și diversificarea serviciilor de calcul și de analiza de date pe care Centrul de Calcul Avansat din IFIN-HH le va oferi comunității științifice naționale și internaționale, contribuind prin aceasta la dezvoltarea sistemului național de CDI și la creșterea vizibilității la nivel european.

Rezultatele prevăzute ale proiectului sunt următoarele:

[8] Cerere de Finantare, proiect CeCBiD-EOSC

Nr.	Rezultat prognozat	Documentul în care a fost raportat
1.	Documentatia de achizitie a serviciilor de consultanta pentru elaborarea documentatiei tehnice necesare echiparii centrului de resurse Cloud si Big Data	RP2
2.	Contract de furnizare a serviciilor de consultanta pentru elaborarea documentatiei tehnice necesare echiparii centrului de resurse Cloud si Big Data	RP2
3.	Documentatie tehnica privind echiparea centrului de resurse Cloud si Big Data	RP3
4.	Documentatia de achizitie a serviciilor de informare si publicitate	RP1
5.	Contract de achizitie servicii de informare si publicitate	RP1
6.	Contracte de achizitie active corporale pentru echiparea centrului de resurse Cloud si Big Data	RP11
7.	Un comunicat de presa publicat la lansarea proiectului	RP1
8.	Un comunicat de presa publicat la finalizarea proiectului	RP13
9.	Pagina web a proiectului, publicata la adresa http://cecbid-eosc.nipne.ro	RP1
10.	Materiale de informare si publicitate (roll-up-uri, afise, rame afis, mape, banner).	RP1
11.	Sistem de procesare de date achizitionat si instalat.	RP13
12.	Sistem de stocare de date achizitionat si instalat.	RP13
13.	Switch pentru interconectarea sistemelor hardware achizitionat si instalat.	RP13
14.	Instalatie de climatizare achizitionata si instalata.	RP13
15.	Sistem UPS achizitionat si instalat.	RP13
16.	Rack-uri pentru gzduirea echipamentelor IT achizitionate si instalate.	RP13
17.	Tablouri electrice si retea electrica achizitionate si instalate.	RP13
18.	Servicii si aplicatii informatice pentru administrarea si monitorizarea centrului CLOUDIFIN, precum si pentru asigurarea accesului utilizatorilor.	RP13
19.	Interfata de acces al utilizatorilor la resurse oferite de centre Cloud multiple. Manual de utilizare.	RP13
20.	Servicii si aplicatii informatice pentru satisfacerea cerintelor IT din faza operationala initiala a proiectului ELI-NP. Raport stiintific si tehnic	RP13
21.	Servicii si aplicatii informatice pentru suportul activitatii de modelare si simulare a nanostructurilor complexe	RP13
22.	Servicii si aplicatii informatice pentru suportul activitatii de analiza a datelor de secventiere de noua generatie	RP13
23.	Studiu privind performantele centrului Cloud si Big Data. Manual de management	RP13

24.	Lucrări și comunicări științifice	RP13
25.	Fișe de post	RP1-13
26.	Documente de raportare (rapoarte de activitate / de progres; procese verbale ale intalnirilor echipei de management a proiectului)	RP1-13
27.	Cereri de plată/rambursare	RP4-13
28.	Raport de audit final al proiectului.	-

Proiectul CeCBiD-EOSC a demarat la data semnării contractului de finațare de catre Ministerul Educației și Cercetării (19.05.2020), iar finalizarea lui este planificată pentru luna iulie 2023.

Proiectul CeCBiD-EOSC este cofinanțat din Fondul European de Dezvoltare Regională (FEDR) în baza contractului de finanțare încheiat cu Ministerul Educației și Cercetării în calitate de Organism Intermediar, în numele și pentru Ministerul Fondurilor Europene în calitate de Autoritate de Management.

Cuprins

Rezumat

Scopul livrabilului

Scopul acestui raport este de a prezenta rezultatele obținute în cadrul Subactivității 2.5 a proiectului CeCBiD-EOSC, privind dezvoltarea software pentru optimizarea analizei datelor de secventiere de noua generatie (NGS) si implementarea platformei de servicii de analiza NGS.

Impact

Workflow-urile dezvoltate in cadrul Subactivitatii 2.5 descriu o solutie originala de automatizare a serviciilor de analiza a datelor de secventiere de noua generatie, care accelereaza procesele de descoperire a diferitelor variante alelice din secvente genomice umane, contribuind la diagnosticarea mai rapida a unor afectiuni complexe precum diabetul, cancerul, boli neurologice, etc. Totodata, platforma software implementata deschide perspectiva dezvoltarii si consolidarii cooperarii cu cercetarea din bioinformatica si cu mediul privat intr-un domeniu emergent si de interes economic, un exemplu in acest sens fiind colaborarea cu SC Genetic Lab SRL.

Conținutul Raportului Tehnic

In Introducere se motiveaza necesitatea analizei secventelor genomice obtinute prin metoda secventierii de noua generatie si se trec in revista obiectivele si rezultatele preconizate ale Subactivitatii 2.5. Proiectarea si testarea a cinci fluxuri de lucru de analiza NGS este prezentata in Cap. 2. Sunt descrise in detaliu 2 fluxuri de lucru optimizate si validate riguros pentru analiza liniei germinative a datelor NGS umane si 3 fluxuri de lucru pentru analiza somatica din probe de secventiere normale si tumorale. Modul de validare a fluxurilor de lucru este prezentat in Sectiunea 2.3, iar comparatia rezultatelor rularii programelor pe infrastructura CPU si, respectiv, GPU, sunt prezentate in Sectiunea 2.4. In Cap. 4 se prezinta activitatile de optimizare si de integrare a fluxurilor de lucru in serviciile de analiza NGS care urmează sa fie puse la dispoziția utilizatorilor in Centrul de Resurse Cloud și Big Data. Platforma software de furnizare a acestor servicii si interfata grafica a utilizatorilor realizate in cadrul CCBD sunt descrise in Cap. 5.

Concluziile Raportului Tehnic

Subactivitatea 2.5 se încheie cu realizarea rezultatului planificat nr. 22 al proiectului - *"Servicii si aplicatii informatice pentru suportul activitatii de analiza a datelor de secventiere de noua generatie. Raport stiintific si tehnic".*

Prin implementarea platformei de analiza a datelor de secventiere de noua generatie se contribuie la indeplinirea urmatoarelor obiective ale proiectului: O4. *Asigurarea conditiilor de sustinere informationala in tehnologie Cloud si Big Data a participarii institutului la colaborari internationale de anvergura, precum si a noilor optiuni strategice privind angajarea in directii de cercetare emergente din spatiul stiintific international, cu relevanta socio-economica deosebita*; O5. *Dezvoltarea si implementarea de servicii si aplicatii informatice pentru administrarea si functionarea centrului de resurse, care utilizeaza tehnologii Cloud si Big Data pentru ... analiza datelor de secventiere de noua generatie*.

De asemenea, prin programarea aplicațiilor de analiza a datelor NGS, Subactivitatea 2.5 a contribuit la îndeplinirea indicatorului suplimentar de realizare "*Aplicații dezvoltate folosind tehnici pentru infrastructuri masive de date (Big Data)*" din Cererea de Finanțare.

1. Introducere

1.1 Context general și necesitate

Obiectivul specific O5 al proiectului include *Dezvoltarea si implementarea de servicii si aplicatii informatice pentru administrarea si functionarea centrului de resurse, care utilizeaza tehnologii Cloud si Big Data pentru analiza datelor de secventiere de noua generatie*.

Analiza secventelor genomice umane (exom sau genom intreg), obtinute prin metoda secventierii de noua genereatie (*New Generation Sequencing* - NGS) este utilizata in vederea localizarii atat a variantelor alelice obisnuite dar si a acelora rar intalnite care pot afecta semnificativ determinismul genetic al diferitelor afectiuni complexe precum diabetul, cancerul, boli neurologice, etc. Prin intermediul genotiparii de mare capacitate la nivelul intregului genom uman s-au identificat mai mult de 2.600 alele cu risc obisnuit, asociate cu peste 350 trasaturi complexe. Progresul tehnologic din domeniul secventierii de noua generatie permite secventierea genomului uman la costuri din ce in ce mai reduse si va facilita identificarea de noi variante rare asociate cu diferite profile patogenice, cavasispecii, sau a mutatiilor *de novo*.

Desi secventierea genomica este din ce in ce mai accesibila atat cercetatorilor cat si beneficiarilor privati, metodele de analiza bioinformatica - mai precis asamblarea genomurilor foarte mari din ampliconi foarte scurti si post-procesarea ulterioara - raman in continuare o provocare datorita cantitatii enorme de date a caror prelucrare prin metode conventionale necesita timp de calcul mare. Acest inconvenient afecteaza atat viteza de progres a cunoasterii stiintifice cat si cea a unor activitati cu aplicabilitate economica, cum este diagnosticarea genetica din probe recoltate de la pacienti/clienti. La fel cum orice metoda de laborator necesita optimizare, testare si validare experimentala, asa si analiza bioinformatica necesita selectia programelor de asamblare (CABOG, SGA, SOAPdenovo) cu rata de eroare scazuta, selectia parametrilor optimi de asamblare pentru tipul de date analizate, precum si validarea metodei de asamblare (e.g. cromozomul 14 uman) conform GAGE (*Genome Assembly Gold-Standard Evauations*).

Rezultatele analizei primare a datelor NGS sunt exemple tipice de Big Data, carora li se pot aplica tehnici avansate specifice pentru sistematizare, data mining, cautari, etc. Acestea au fost implementate pe platforma CLOUDIFIN in vederea accelerarii proceselor de descoperire sau diagnosticare.

1.2 Obiectivele Subactivității 2.5

Obiectivele specifice ale Subactivității 2.5 sunt următoarele [8]:

1. *Instalarea sistemelor de management al fluxurilor de lucru Apache Taverna si Galaxy*
2. *Testarea aplicatiilor care folosesc tehnici pentru infrastructuri masive de date (Big Data) pentru analiza bioinformatica a datelor NGS*
3. *Instalarea si testarea apliatiilor bioinformatice de analiza NGS care ruleaza pe acceleratoare grafice*
4. *Proiectarea si testarea fluxurilor de lucru bioinformatice pentru asamblarea secventelor de novo si pe referinte, genotipare, detectie SNP, detectie indel, etc.*
5. *Validarea acuratetii fluxurilor de lucru bioinformatice pentru analiza datelor NGS*

1.3 Rezultate preconizate

- 22 - Servicii si aplicatii informatice pentru suportul activitatii de analiza a datelor de secventiere de noua generatie. Raport stiintific si tehnic.

2. Proiectarea si testarea fluxurilor de lucru

Seturile de date genomice umane au in general doua proveniențe: germinativa si somatica. In timp ce seturile de date obținute din linie genetica sunt întrebuințate in general pentru studii genomice si pentru identificarea alelelor implicate in etiologia maladiilor ereditare, iar analiza liniei somatice se realizează cu scopul identificării modificărilor genetice implicate în diferite tipuri de neoplazii. Analiza bioinformatică a liniei genetice somatice presupune analiza comparativă a două probe de secventiere: una obținută din celule normale si cealaltă din celule tumorale. Fluxurile de lucru dezvoltate in cadrul proiectului reflecta aceste diferențe. La finalul proiectului s-au obținut 2 fluxuri de lucru optimizate si validate riguros pentru analiza liniei germinative a datelor NGS umane si 3 fluxuri de lucru pentru analiza somatica din probe de secventiere normale si tumorale. Ambele tipuri de fluxuri de lucru pot fi utilizate pentru analiza seturilor de date WGS/WES de secventiere Illumina obținute din probe clinice.

Fluxurile de lucru utilizate in vederea automatizării procedurilor de analiza bioinformatica a datelor NGS brute sunt bazate pe sistemului de management al fluxurilor de lucru Apache Taverna [1]. A fost ales sistemul Taverna datorita experienței anterioare și pentru că păstrează posibilitatea de rulare atât pe infrastructura HPC cât și în Cloud. Fluxurile de lucru Taverna au o structura generala alcătuita din: 1) porturi de intrare prin care se specifica datele NGS brute si parametri esențiali; 2) module folosite pentru execuția programelor si 3) porturi de ieșire folosite pentru diagnostic si validarea execuției fluxului de lucru. Fluxurile de lucru prezinta o construcție modulara, iar fiecare modul primește date de intrare individual, astfel încât să permită reluarea fluxului exact din punctul unde a survenit o eroare, fără a necesita rularea modulelor precedente. Toate fluxurile de lucru rezultate au fost testate pe un server HPC dotat cu două procesoare Intel Xeon Gold 5120 cu 14 nuclee, 128 GB memorie RAM, și sistem de operare CentOS 7. Aceleasi fluxuri de lucru au fost rulate in paralel pe o mașină virtuală din clusterul CLOUDIFIN dotată cu 24 de nuclee CPU (Intel Xeon Gold 5120), 100 GB memorie RAM, având instalat același sistem de operare (CentOS 7). Ambele sisteme de calcul au avut instalat un accelerator grafic NVIDA Tesla K80.

Primul flux de lucru implementat si testat in cadrul proiectului s-a axat pe descoperirea, adnotarea și interpretarea variantelor umane scurte: single-nucleotide polymorphism (SNP) și inserții/deletii (INDEL), bazat in principal pe pachetului software Genome Analysis Toolkit (GATK) [2]. GATK este unul dintre cele mai utilizate instrumente software de identificare a variantelor genetice i.e. identificarea cu precizie suficient de buna a diferențelor dintre citirile analizate si genomul de referința, care sunt determinate fie de variații genetice reale fie de erori. Pentru obținerea acestor performante, pachetul GATK utilizează metode statistice avansate precum regresia logistica, clasificatorul bayesian naiv si modelul Markov ascuns. Pe scurt, citirile NGS brute au fost aliniate pe o versiune a genomului uman de referință (GRCH37 / GRCH38) cu ajutorul programului BWA-MEM [3]. Fisirele FASTQ aliniate au fost convertite in fișiere BAM, care mai departe au fost sortate, indexate, iar citirile duplicatele au fost marcate. Fișierele BAM au fost rafinate cu ajutorul modulului Base Quality Score Recalibration (BQSR) cu parametrii standard. Identificarea efectivă a variantelor scurte a fost realizata cu ajutorul modulului HaplotypeCaller. Pentru eficientizarea procesării datelor aliniate, au fost împărțite in 8 segmente care au cuprins 2-3 cromozomi si care au fost procesate mai departe in paralel, dar au fost combinate in final intr-un singur fișier VCF. In cele din urma, fișierul VCF rezultat a fost filtrat cu ajutorul modulului Variant Quality Score Recalibration (VQSR) folosind diferite seturile de training pentru filtrarea SNP si INDEL.

Pe parcursul proiectului s-au efectuate o serie de actualizări si optimizări asupra fluxului de lucru dezvoltat pentru identificarea variantelor scurte din linie genetică germinativă (GATK). GATK este limitat in ceea ce privește managementul memoriei si eficienta rulării i.e. module importante sunt încă limitate la rularea pe un singur nucleu CPU sau doar pe câteva nuclee. Din cest motiv este esențială organizarea eficienta a datelor brute aliniate si paralelizarea acestora. Optimizarea modului in care sunt procesate datele a inclus determinarea numărului optim de segmente in care pot fi împărțite fișierele BAM, rezultate in urma alinierii, care pot fi procesate

cat mai eficient pe numărul de nuclee CPU disponibile pe infrastructura HPC sau pe cloud. O alta modificare a fluxului de lucru a fost posibilitatea de selectare a versiunii de genom de referință GRCh38 sau GRCh37. A fost adăugată versiunea mai veche (GRCh37) deoarece este încă cea mai folosită versiune. Ultima versiune a genomului de referință (GRCh38) tine mult mai bine cont de complexitatea genomului uman si include loci alternativi i.e. multiple versiuni distincte ale regiunilor complexe ale genomului. Al doilea motiv pentru includerea genomului GRCh37 in fluxul de lucru fiind faptul că majoritatea programelor folosite pentru evaluarea specificității si sensibilității metodei utilizate pentru determinate variantele scurte încă nu iau in considerare contig-urile alternative ale versiunii GRCh38.

Tot in vederea identificarii variantelor scurte din linie genetică germinativa a mai fost implementat un flux de lucru bazat pe pachetului software DeepVariant [4], ca alternativa la fluxul de lucru GATK. DeepVariant utilizează un algoritm *deep learning* bazat pe Google TensorFlow, beneficiind astfel de îmbunătățirea eficientei de procesare, datorită utilizării acceleratoarelor grafice (GPU). Ambele fluxuri de lucru au primit îmbunătățiri, precum posibilitatea de a descărca seturi de date NGS (WGS/WES) direct din European Nucleotide Archive pe baza codului de acces, dar si posibilitatea de a selecta diferite programe de aliniere in etapa de pre-procesare.

Identificarea mutațiilor somatice nu ar trebui sa fie dificila daca se compara seturile de datele NGS normale cu cele tumorale, deoarece modificările asociate diferitelor tipuri de cancer sunt prezente doar în proba tumorală si nu in alela de referință din proba normala. Dar atât factorii biologici cât si tehnologici, care includ eterogenitatea intra-tumorală, contaminarea probelor, incertitudinile de citirea a bazelor și alinierea citirilor de secvențiere, cresc semnificativ gradul de dificultate al identificării mutațiilor somatice. Studiile diferitelor populații clonale si sub-clonale au dezvăluit că celulele tumorale variază în modul în care sunt anormale, iar unele mutații pot fi observate doar la o mică parte din celulele tumorale de la un pacient. De asemenea, obținerea unor probe tumorale si normale pure, cu ajutorul tehnologiilor actuale, este aproape imposibilă. Contaminarea probelor duce la subestimarea fracțiilor variantelor alelice (VAF) in probele tumorale, respectiv supraestimare in probele normale. Fluxurile de lucru implementate si testate in cadrul proiectului includ pași de procesare pentru reducerea gradului de incertitudine in identificarea mutațiilor somatice, respectiv analiza CNV. In mare majoritate a cazurilor, celulele umane normale conțin doar doua copii ale genelor, dar in anumite situații numărul copiilor poate creste. Acest efect se numește *Copy Number Variants* (CNV). Efectul se poate manifesta atât pe linie germinativă cât si somatică. Una din situațiile in care se întâlnește CNV somatic este in cancer. Majoritatea tumorilor conțin CNV, dar impactul acestora asupra evoluției clinice a cancerului este încă slab înțeles.

S-au dezvoltat si validat doua fluxului de lucru pentru identificarea variantelor scurte din linie genetica somatica, bazate pe programele GATK MuTect2 [5] si Strelka2 [6]. Fluxul de lucru pentru identificarea CNV (*Copy Number Variants*) este bazat tot pe pachetul GATK. Etapa de pre-analiza a fluxurilor de lucru pentru identificarea variantelor scurte (SNP si INDEL) si CNV somatice prezint aceleași module in vederea descărcării seturilor de date, alinierii, sortării, indexării, etc. Pe scurt, structura fluxului de lucru GATK MuTect: citirile NGS brute au fost aliniate pe o versiune a genomului uman de referință (GRCH37 / GRCH38) a cu ajutorul programului BWA-MEM. Citirile aliniate au fost convertite in fișiere BAM, iar mai departe au fost sortate si indexate cu ajutorul programului SAMtools [7] iar duplicatele au fost marcate (GATK MarkDuplicates). Identificarea efectivă a variantelor scurte a fost realizata cu ajutorul modulului GATK MuTect2. Variantele rezultate au fost mai departe filtrate cu ajutorul modulului GATK FilterMutectCalls pentru excluderea fals pozitivelor. In final, variantele au fost adnotate cu ajutorul modulului GATK Funcotator. Variantele finale sunt scrise in format VCF multi-probă i.e. sunt înregistrate variantele descoperite in proba normală față de genomul de referință, respectiv variantele descoperite in proba tumorala și care nu se regăsesc în proba normală. Variantele filtrate si adnotate tumorale au fost extrase cu ajutorul programului BCFtools [8].

2.1 Identificarea variantelor scurte (SNP si INDEL) din linie genetică germinativa

In primele etape ale proiectului a fost implementat și testat fluxul de lucru bazat pe pachetul software Genome Analysis Toolkit (GATK) pentru descoperirea, adnotarea și interpretarea variantelor umane scurte SNP si INDEL. Fluxul de lucru este bazat pe setul de recomandări GATK de bune practici pentru analiza datelor de secventiere cu rata mare de transfer (HTS), pornind de la datele de brute de secventiere (NGS) în format FASTQ sau uBAM (unmapped). Fluxul conține 3 faze de procesare: (1) pre-procesarea datelor brute; (2) descoperirea variantelor din linia germinativă; (3) analiza preliminara - adnotarea avansată și filtrarea variantelor (Fig. 2.1).

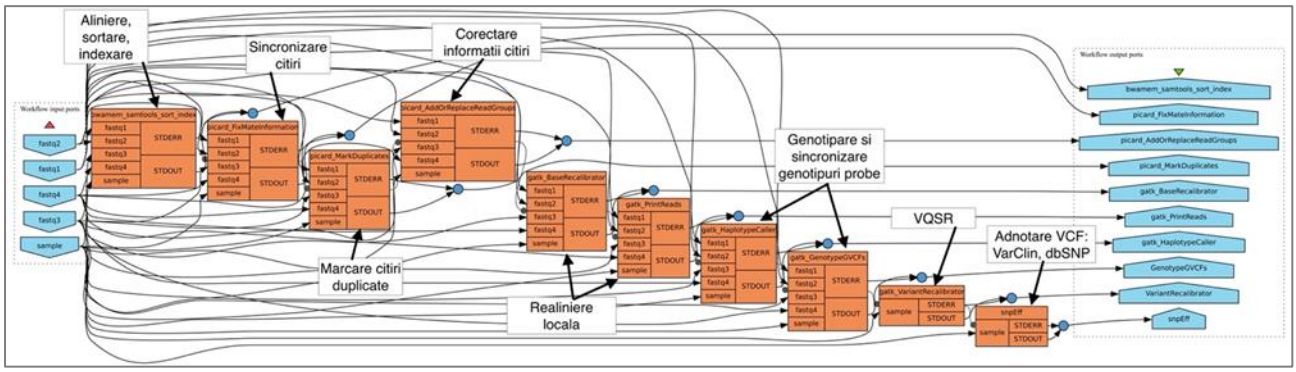


Figura 2.1 Diagrama fluxului de lucru Taverna pentru pre-procesarea datelor brute NGS, descoperirea variantelor genetice și adnotarea variantelor

Primul modul al fluxului de lucru folosește programul BWA (Burrows-Wheeler Alignment cu algoritmul BWA-MEM) [3] pentru alinierea perechilor de citiri (forward și revers) pe „genomul de referință" GRCh38DH. Acest genom include un set de contig-uri reprezentative pentru fiecare cromozom: 1-22 (chr1-22), X (chrX), Y (chrY) și genomul mitocondrial (chrM), dar și un set de contig-uri alternative, asamblări alternative și loci alternativi pentru a reprezenta cele mai des întâlnite variații complexe (e.g. loci HLA), și care permit asamblarea corectă a haplotipurilor divergente. Pe lângă regiunile alternative, mai este inclus un contig de „captura" care mapează secvențele endogene ale virusului Epstein-Barr (infectează limfocitele B în ~90% din populația mondială). În acest prim modul sunt incluse etapele de sortare și indexare a fișierelor BAM, recent mapate, realizate cu ajutorul pachetului software SAMtools [7] - necesare procesării ulterioare și vizualizării asamblărilor.

In urma alinierii și sortării citirilor perechi, este necesara verificarea și eventual sincronizarea citirilor care nu se suprapun perfect cu citirea complementara, forward și reverse, care compun fragmentele (inserții) de secvențiere Illumina. Următorul pas al fluxului de lucru este reprezentat de marcarea citirilor duplicate, care pot surveni în etapa de pregătire a librăriilor prin amplificarea PCR, sau pot apărea ca urmare a interpretării incorecte a unui cluster de amplificare ca multiple clustere. Citirile neîmperecheate cu citirea complementara sunt marcate tot ca duplicate și vor fi îndepărtate. Următorul modul al fazei de pre-procesare nu este specificat în ghidul de bune practici GATK,dar presupune adăugarea/corectarea informațiilor suplimentare (e.g. specificarea platformei de secvențiere, corectare grupurilor de citire, etc.) conținute în mod normal de fișierele BAM, care sunt necesare pentru faza de descoperire a variantelor. Modulele 2-4 ale fluxului de lucru utilizează pachetul software Picard [9] pentru pre-procesarea datelor. Algoritmul de aliniere BWA-MEM este foarte eficient, dar prezinta totodată și o rată ridicată de aliniere incorectă a fragmentelor, în special dacă variantele sunt localizate la capetele fragmentelor. Pentru a corecta erorile de aliniere introduse de BWA, este necesara realinierea locala ținând cont de SNP/INDEL cunoscute (e.g. dbSNP138, hg38_known_indels, 1000g_gold_standard_indels).

In etapa de identificare a variantelor primare ale liniei germinative, ghidul de bune practici GATK recomandă utilizarea subprogramului complex HaplotypeCaller [10]. Programul identifică regiunile active i.e. care prezinta un nivel semnificativ de variație; construiește grafice tip De Bruijn pe baza cărora reasamblează regiunile active și determina haplotipurile posibile; folosește algoritmul PairHMM pentru a produce o matrice de probabilități ale haplotipurilor care sunt apoi utilizate pentru a determina alelele probabile pentru fiecare regiune activa; în final este determinat genotipul prin aplicare teoremei lui Bayes. Programul HaplotypeCaller este eficient în identificarea variantelor din regiunile cu un grad ridicat complexitate e.g. conțin diferite tipuri de variante apropiate una de alta.

Pentru determinarea variantelor genetice finale, este necesara filtrarea variantelor descoperite inițial, in funcție de diferite caracteristici (e.g. complexitatea regiunii, gradul de acoperire al alelei, raportul de acoperire dintre citirile forward/revers, etc.), ghidata cu ajutorul variantelor genetice cu grad ridicat de caracterizare (1000 Genomes și hapmap) pentru a exclude variantele fals pozitive. Acest proces se numește recalibrarea scorurilor de calitate ale variantelor (VQSR), și folosește un algoritm ML (Machine Learning) în care variantele genetice validate sunt utilizate pentru obținerea setului de antrenare. Acest set integrează 5-8 parametrii, din profilurile de adnotare, pentru alcătuirea modelului de recunoaștere a variantelor adevărate dintre cele false din setul de testare. Metoda VQSR se aplica separat pentru SNP și INDEL, rezultând doua fișiere VCF (variant call format).

Fișierele VCF obținute în etapa de descoperire a variantelor genetice sunt supuse mai departe analizei preliminare - adnotarea avansata și selecția variantelor relevante pentru fiecare tip de analiza medicală / studiu clinic. Adnotarea variantelor include date elementare legate de efectele biologice: gena afectată; natura regiunii afectate - codanta sau necodantă; mutație sinonimă sau ne-sinonimă (*missense* sau *non-sense*). Ultimul modul al fluxului de lucru este dedicat adnotării și selecției variantelor genetice cu relevanta clinică, utilizând programele SnpEff [11] și SnpSift [12]. Pe lângă adnotarea variantelor genetice, SnpEff mai are și funcția de predicție a efectelor biologice ale variantelor folosind o baza de date proprie de „genomuri de referință”. Programul SnpSift permite adnotarea variantelor genetice cu câmpuri din alte fișiere VCF (e.g. dbSnp, ClinVar, ExAC, etc.). Variantele genetice (> 500.000 obținute de la >1200 pacienți) puse la dispoziție prin intermediul bazei de date ClinVar [13] au semnificație clinica e.g. benignă / probabil benignă, patogenică / probabil patogenică, factor de risc, răspuns la medicament, etc.

Programul DeepVariant înlocuiește regresia logistică pentru modelarea erorile de bază, modele Markov ascunse pentru a calcula probabilitățile de citire, si clasificarea Bayes pentru identificarea efectivă a variantelor, utilizate de programul GATK, cu un singur model de *deep learning* (DL). Lanțul de instrumente DeepVariant începe prin identificarea SNP/INDEL din contig-urile aliniate la genomul de referință cu sensibilitate ridicată, dar cu specificitate scăzută, folosind tehnici standard de procesare algoritmică. Modelul DL folosește arhitectura *Inception* [14]- bazată pe rețele neurale convoluționale - ce emite probabilități pentru fiecare dintre cele trei genotipuri diploide (homozigot raportat la referință, heterozigot si homozigot alternativ) pentru fiecare locus. Modelul este antrenat cu ajutorul genotipurilor adevărate etichetate, care poate fi stocat si aplicat pe probe de secvențiere noi.

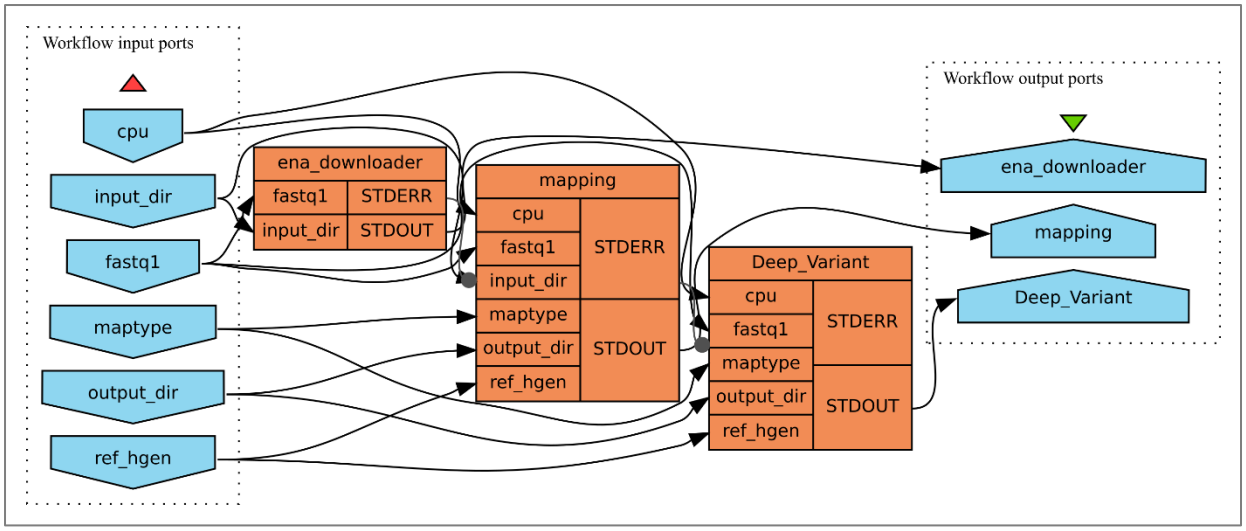


Figura 2.2 *Diagrama fluxului de lucru Apache Taverna pentru identificarea variantelor scurte SNP si INDEL, din linie genetica germinativa umana cu ajutorul programului DeepVariant.*

Fluxul de lucru DeepVariant este mult mai simplu, fiind format din 3 module: 1) descărcarea seturilor de date din European Nucleotide Archive (ENA); 2) alinierea citirior NGS Illumina la genomurile de referinta umane (GRCh38 si GRCh38/hg19); 3) identificarea propriu-zisa a variantelor scurte din probele genomice WGS (*Whole Genome Seqencing*) si WES (*Whole Exome Seqencing*) cu programul Deep Variant.

2.2 Identificarea variantelor scurte (SNP si INDEL) si CNV din linie genetică somatica

S-a implementat și testat fluxul de lucru bazat pe pachetul software Genome Analysis Toolkit (GATK) pentru descoperirea, adnotarea și interpretarea variantelor SNP/INDEL, precum si pentru determinarea variației numărului de copii (CNV) din linie genetica somatica. Pentru testarea initiala a fluxului de lucru s-au folosit date genomice brute obținute din țesut mamar (proba normale) si tumora mamara (proba tumorala) publicate de către P. Savage et. al [15].

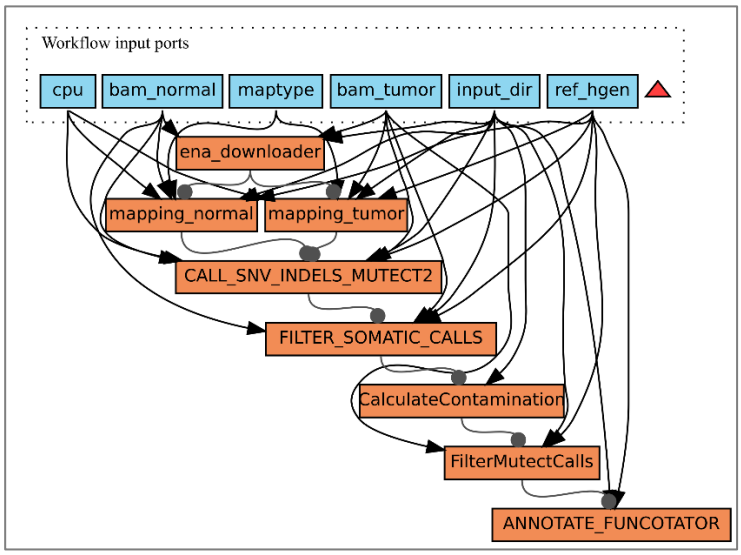


Figura 2.3 *Diagrama fluxului de lucru Apache Taverna pentru identificarea variantelor somatice umane din probe normale si tumorale, cu ajutorul programului Mutect2 (GATK4).*

Identificarea preliminara a mutațiilor somatice, atât din proba tumorala cat si din proba normală, s-a realizat cu ajutorul programului GATK MuTect2 [5]. Pe baza datelor preliminare, programul construiește trei seturi de filtre utilizând: proba normală, care permite excluderea variantelor ce aparțin linei germinative; o baza de date populațională (gnomAD), care documentează cele mai frecvente alele germinative umane; un panel de normali (PoN), care asigură captarea alelelor ramase nedetectate in pașii anteriori. Un panel de normali este realizat din probe genomice integrale (din 10-40 probe) tehnic similare, secventiate pe aceeași platforma, cu același kit de secventiere, aliniate pe același genom, folosind aceeași metodă de analiza NGS. Filtrarea cu ajutorul PoN permite identificare si excluderea artefactelor de citire si mapare, dar si a altor artefacte randomice de secventiere si procesare. Cele trei filtre sunt aplicate in următoarele 3 module ale fluxului de lucru astfel încât parametrii care afectează sensibilitate si specificitatea fiecărui metode de filtrare sa fie ajustați individual (Fig. 2.3). Ultimul modul adnotează variantele somatice identificate cu ajutorul programului Funcotator, care apelează baze de date specializate in cele mai prevalente tipuri de cancer.

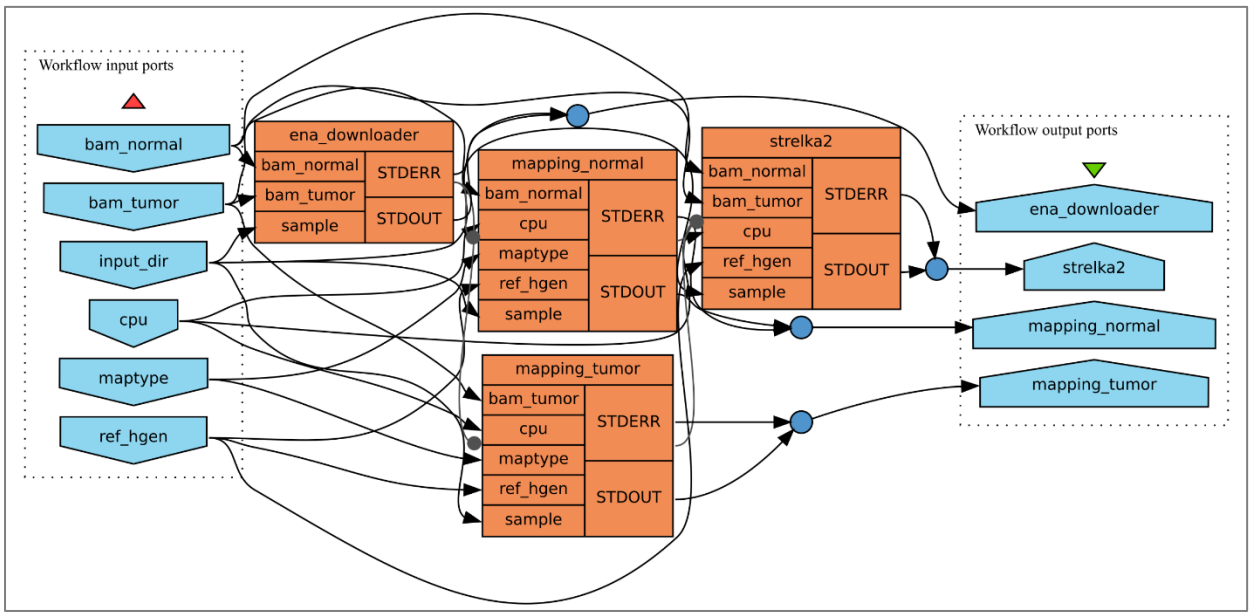


Figura *2.4.* *Diagrama fluxului de lucru Apache Taverna pentru identificarea variantelor somatice umane din probe normale si tumorale, cu ajutorul programului Strelka2*

Procedura folosita de programul Strelka2 pentru identificarea variantelor somatice prezinta unele avantaje fata de programul Mutect2, in special in cazul analizei seturilor de date hibride si sintetice i.e. generate *in silico*. De asemenea, programul Strelka2 este mult mai eficient din punctul de vedere al procesării computaționale, comparativ cu Mutect2. Ambele fluxuri de lucru pot fi utilizate pentru confirmarea modificărilor somatice din probe clinice de secventiere NGS cu acoperire ridicata: >100X.

Primul modul al fluxului de lucru pentru analiza CNV citește datele NGS aliniate ale probelor tumorale/normale, alcătuiește o listă de intervale geonomice corespunzătoare și colectează numărul de citiri per intervale. Intervalele genomice sunt cele determinate cu ajutorul datelor asociate kiturilor de secvențiere NGS, si corespund in general locilor alelici. Următoarele doua module alcătuiesc panelul de normali (PoN) pentru CNV, respectiv reducerea zgomotului citirilor fata de PoN, cu ajutorul analizei componentelor principale. Următorul modul determina raportul dintre alelele de referință si alelele alternative. In final, ultimul modul încorporează raportul copiilor si numărătoarea alelică, si le grupează in segmente continue (cromozomiale) cu ajutorul metodei MCMC (Markov-Chain Monte Carlo).

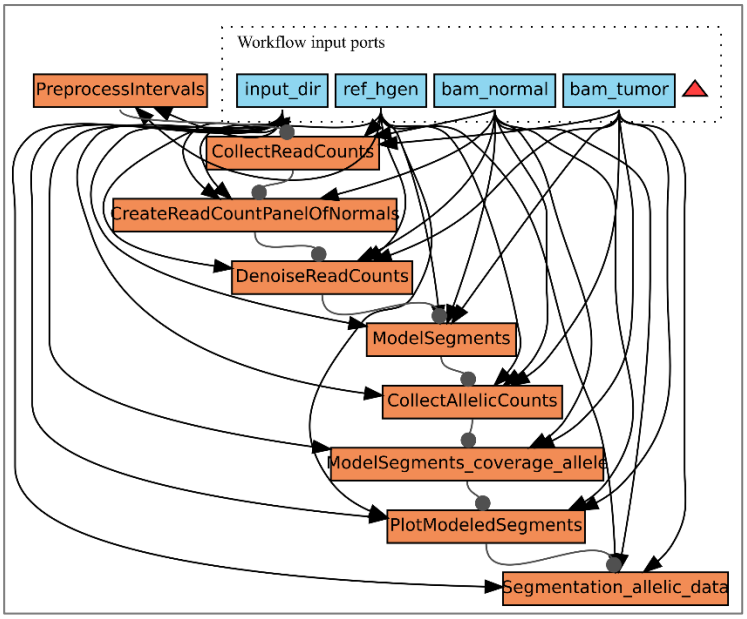


Figura 2.5 *Diagrama fluxului de lucru Apache Taverna pentru identificarea modificărilor genomice CNV (Copy Number Variants) din probele normale si tumorale, cu ajutorul pachetului de programe GATK.*

2.3 Validarea fluxurilor de lucru

2.3.1 Identificarea variantelor scurte din linie germinativa (GATK si DeepVariant)

S-a evaluat impactului modificărilor efectuate asupra fluxului de lucru in ceea ce privește consistenta rezultatelor, precum și validarea rezultatelor finale i.e. compararea fișierelor VCF brute / filtrate cu VQRS cu date genomice foarte bine caracterizate e.g. NA12878 considerat „standard de aur”. Variantele genetice pot fi reprezentate în diferite modalități in cadrul formatului VCF, iar daca se compară direct datele din doua seturi, este posibil ca majoritatea diferențelor sa fie date de diferitele reprezentări ale aceleiași variante. Mai mult de atât, definițiile parametrilor de performantă esențiali pentru validarea metodelor, precum adevărat pozitiv (TP), fals pozitiv (FP), si fals negativ (FN), nu sunt încă standardizate. In cele din urma, performantele metodelor variază in funcție de tipurile de variante si de complexitatea regiunilor genomice utilizate pentru comparație. Parametrii esențiali pentru validarea analitica a variantelor identificate sunt sensibilitatea - abilitatea de a detecta variante care se cunosc ca sunt prezente sau absenta variantelor fals negative (FN) denumita „Recall” si specificitatea - abilitatea de a identifica corect absenta variantelor sau excluderea prezentei fals pozitivelor (FP) denumita „Precision”. Cei doi parametrii „Recall” si „Precision” sunt combinați intr-unul singur parametru - „F1_score”:

$$Recall = TP/(TP+FN)$$

$$Precision = TP/(TP+FP)$$

$$F1\_Score = 2 * Precision * Recall/(Precision + Recall)$$

Setul de variante genetice de referință pentru genomul NA12878 este publicat de Genome in a Bottle (GIAB) [16]. Acest set de variante genetice caracterizează cu un grad ridicat de încredere un genotip uman pentru 78% din bazele nucleotidice cu informații de secvența (i.e. baze diferite de „N”) din genomul de referință GRCh37. Datele de intrare pentru comparația rezultatelor constau din fișierele VCF brute si cele rafinate cu VQSR, atât pentru SNP cat si pentru INDEL, care sunt comparate individual. Pe lângă acestea fiind necesar si setul de variante genetice de

referință (VCF) si un set corespunzător de regiuni de încredere pentru acestea in format BED. Regiunile de încredere indică locațiile geonomice folosite pentru comparație. Variantele din setul testat care nu se potrivesc cu variantele din referință sunt considerate FP, iar variantele de referință care nu se potrivesc cu niciuna din setul de test sunt considerate FN. Setul de regiuni de încredere a fost dezvoltat de Platinum Genomes [17].

Un set adițional de variante de referință a fost dezvoltat ceva mai recent dintr-o mixtură a doua mole hidatiforme (i.e. sarcină molara), respectiv liniile celulare hidatiforme CHM1 și CHM13, rezultând un genom "diploid sintetic". Deși variantele din aceste doua seturi de referință reprezintă scenarii reale, numărul de variante adevărate este de obicei necunoscut, complicând utilizarea acestora pentru evaluarea acurateței (i.e. cat de aproape este setul definit de variante adevărate de peisajul mutațional „adevărat”). In schimb, variantele de referință obținute din genotipului "diploid-sintetic" pot fi utilizate pentru generarea si evaluarea datelor de secvențiere NGS obținute in silico i.e. generarea de variante intr-un scenariu controlat cu o rata mutațională predefinită, complementând validarea cu date reale. In acest studiu, datele de secvențiere NGS si variantele de referință pentru genomul „diploid sintetic” au fost folosite pentru evaluarea acurateței fluxului de lucru in comparație cu date din literatura. Datele de secvențiere geonomica integrala (WGS) folosite pentru validare au fost publicate de către Supernat et al. [18] si au fost descărcate din baza de date NCBI SRA (cod de acces SRR6794144). S-a folosit SRA Toolkit pentru obținerea perechii de fișiere FASTQ nealiniate. Datele WGS brute aparținând genomului "diploid sintetic" au obținute prin secvențierea unui amestec (raport 1:1) al liniilor celulare CHM1 (cod de acces ENA: SAMN02743421) si CHM13 (cod de acces ENA: SAMN03255769). Au fost descărcate doua seturi de date rezultate din doua secvențieri replicate, realizate de doua echipe independente, având codurile de acces ENA: ERR1341793 si ERR1341796. Cele doua perechi de fișare FASTQ aparținând genomului "diploid sintetic" au fost analizate cu noua versiune a fluxului de lucru (GARK v4.2.0) folosind versiunea GRCH37 a genomului de referință, iar evaluarea rezultatelor s-a realizat prin comparația cu variantele de referință si regiunile de încredere corespunzătoare (fișierele full.37d5.vcf sid full.37d5.bed incluse in CHM-eval [19]. Cele doua fișiere FASTQ aparținând genomului NA12878 au fost analizate cu ambele variante ale fluxului de lucru (i.e. GATK v3.8 si GATK v.4.2.0) și cu cele doua variante ale genomului de referință (GRCH37 si GRCH38). Fișierele rezultate au fost comparate cu versiunile corespunzătoare ale variantelor de referință si regiunile de încredere. Evaluarea parametrilor de performanta a fost realizata cu ajutorul programului hap.py v0.3.14 [20] cu motorul de analiza vcfeval.

Indicatorii de performanta obținuți in cazul analizei probei SRR6794144 in diferite situații (i.e. utilizarea genomului de referință GRCh38 / GRCh37; versiunii GATK 3.8 / 4.2.0) au fost relativ apropiați. Au fost identificate aproximativ 3.9 milioane de SNP si aproximativ 800 mii de INDEL. Numărul total al variantelor cunoscute pentru GRCh38 fiind de 3.54 milioane (3.04 mil. SNP si 0.499 mil. INDEL), respectiv 3.69 mil. (3.21 mil SNP si 0.48 mil INDEL) cu versiunea GRCh37. Diferențele fată de setul de testare sunt considerate variante necunoscute (0.6 - 0.8 mil. SNP si 0.3 mil. INDEL). Diferențele dintre indicatorii de performanta in cazul utilizării celor doua versiuni ale genomului de referință (Fig. 2.6) sunt reduse, atât pentru sensibilitate cat pentru specificitate, dar cu un ușor avantaj in favoarea versiunii GRCh37. Calculul ariei de sub curba ROC arata situația la nivel global și indică reducerea sensibilității de la 98.5 % (GRCh37) la 96 % cu GATK v.3.8 si la 96.4 % cu GATK v4.2.0 in cazul utilizării versiunii GRCh38 a genomului de referință. Analiza fișierelor VCF filtrate cu ajutorul modulului VQSR indică scarificarea sensibilității in favoarea specificității: 94 % pentru GRCh37 si 91.96 % (v4.2.0) si 91.76 % (v3.8) pentru GRCh38. Acești indicatori, precum si analiza ariei de sub curba ROC indică îmbunătățirea marginală a sensibilității si specificității in cazul utilizării GATK v4.2.0. Cea mai mare îmbunătățire este reprezentată de creșterea sensibilității (2 %), ce se observa in cazul utilizării versiunii GRCh37 a genomului de referință, care determina identificarea suplimentară a aproximativ 150 mii de SNP fată de analiza realizata cu versiunea GRCh38.

Metoda ML folosita de modulul VQSR determină reducerea semnificativă (>10 mii) a variantelor fals pozitive, atât in cazul in cazul SNP cât și al INDEL, pentru toate situațiile calculate, dar cu prețul reducerii sensibilității metodei i.e. filtrarea exagerata a variantelor adevărat pozitive. Din păcate, modificarea parametrilor standard recomandați (reducerea parametrului max gaussian)

pentru modulul ApplyVQSR nu a condus la păstrarea echilibrului dintre sensibilitate si specificitate, in ciuda limitării procesului de filtrare. In aceste condiții, pentru majoritatea aplicațiilor fluxului de lucru, filtrarea variantelor cu ajutorul modulului VQSR nu poate fi justificată.

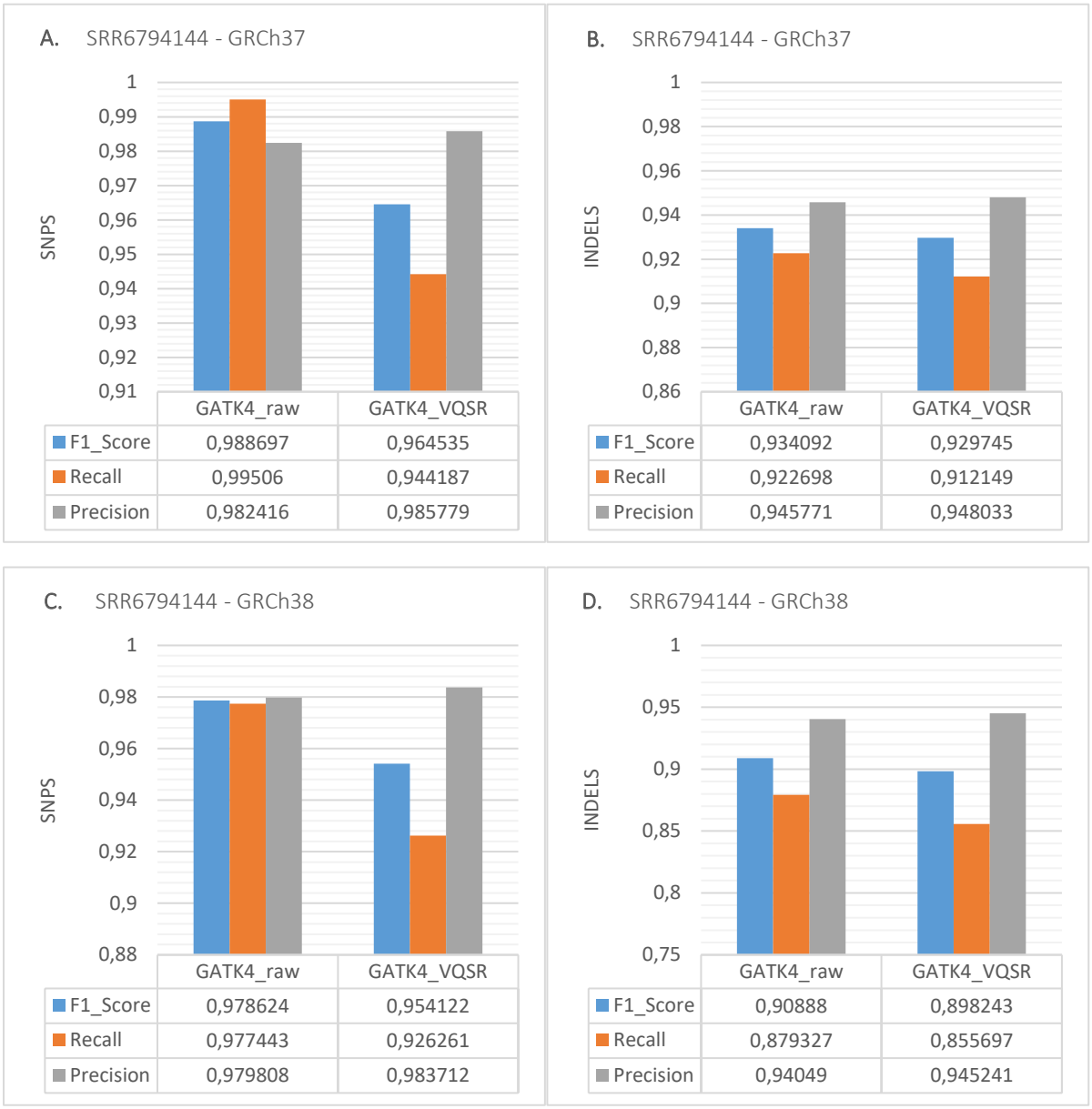


Figura 2.6 Indicatori de performanta pentru proba SRR6794144 (NA12878 / HG001) la nivel de SNP (A) si INDEL (B) analizata cu ajutorul versiunii GRCh37 a genomului de referință; la nivel SNP (C) si INDEL (D) analizata cu ajutorul versiunii GRCh38 a genomului de referință.

Având in vedere ca probele ERR1341796 si ERR1341793 sunt replicate ale genomului "diploid sintetic", analiza acestora a produs indicatori de performanta foarte apropiați, iar micile diferențe se încadrează in marja de eroarea a metodei. Fișierul de referință VCF folosit pentru acesta analiza conține 3.55 mil. SNP si 0.54 mil. INDEL. Analiza comparativă a variantelor celor doua probe replicate a identificat un număr total de 3.18 mil. SNP si 0.48 mil. INDEL. Ca si in cazul variantelor identificate pentru proba SRR6794144, filtrarea acestora cu modulul VQRS determina o ușoară creștere a specificității in detrimentul sensibilității pentru ambele tipuri de variante (Fig.

2.7). Rezultatele obținute pentru validarea fluxului de lucru cu toate probele analizate fiind in concordanță cu datele din literatura [18].

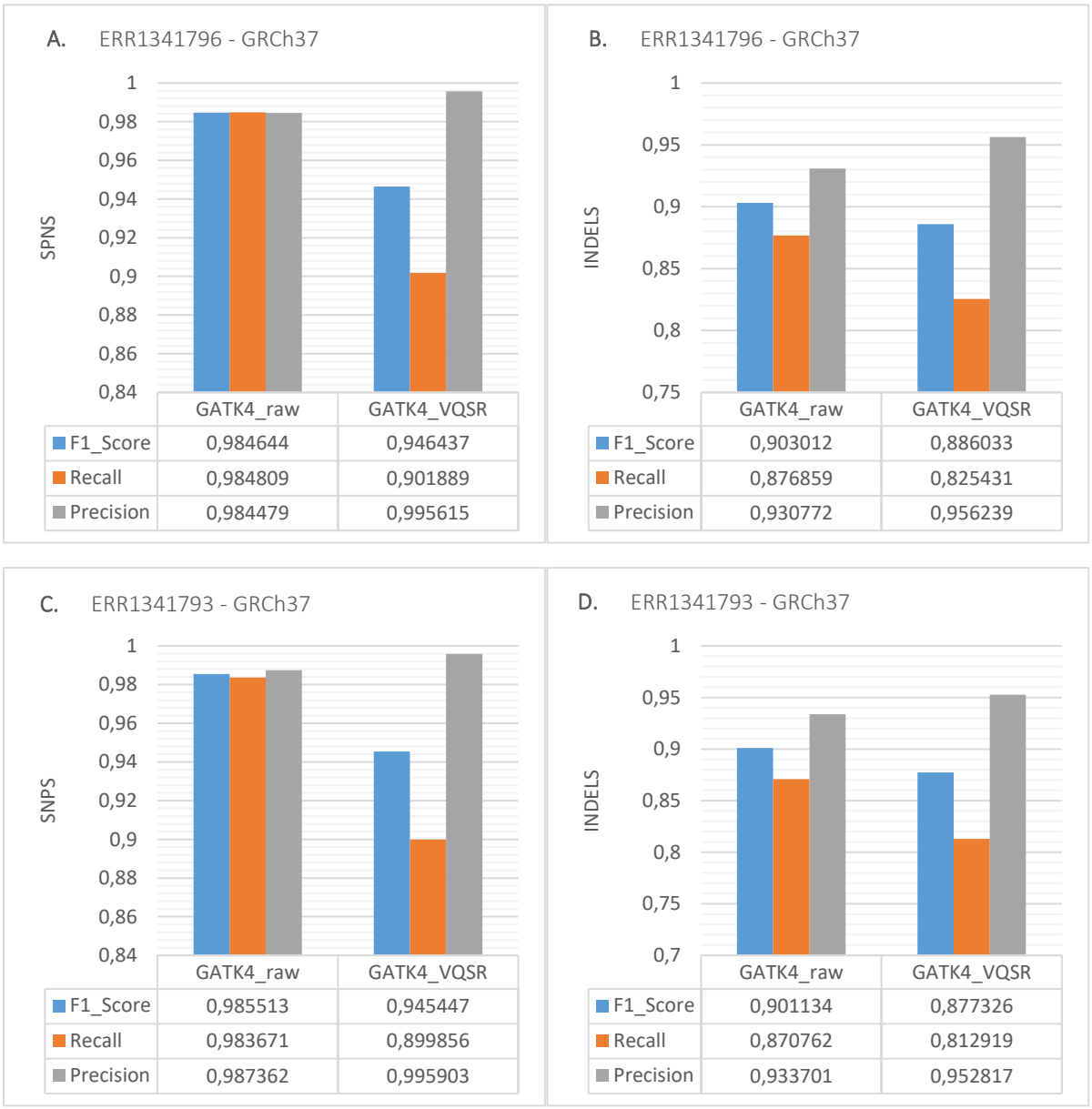


Figura 2.7 Indicatori de performanta pentru proba ERR1341796 la nivel de SNP (A) si INDEL (B); pentru proba ERR1341793 la nivel de SNP (C) si INDEL (D). Ambele probe au fost analizata cu ajutorul versiunii GRCh37 si GATK v4.2.0.

Validarea rezultatelor obținute cu ajutorul programului DeepVariant s-a realizat pe baza genomului HG003 / NA24149, cromozomul 20. Indicatorii de performanță au fost comparați cu cei publicați de dezvoltatorii programului DeepVariant. Cele două seturi de rezultate fiind identice pentru SNP: *Recall* 0.995995, *Precision* 0.999086 si scor F1 0.997538; si pentru INDEL: *Recall* 0.995295, *Precision* 0.998010 si scor F1 0.996651. Mai departe s-a utilizat genomul uman HG001 / NA12878 (SRR6794144) pentru a evalua parametrii de performanță deja obținuți cu fluxul de lucru GATK4, care au fost comparați cu datele obtinute cu DeepVariant. Folosind DeepVariant, s-a analizat o variantă simplă a setului de date (citirile doar indexate și sortate), si o variantă a setului de date pre-procesată complet cu ajutorul pachetul GATK4 (analizată

anterior si cu GATK4 HaplotypeCaller). Din comparația indicatorilor de performanță obținuți pentru cele două programe de analiza, reiese creșterea ușoară a sensibilității și specificității pentru detecția SNP cu programul DeepVariant față de GATK4 (Fig. 2.6A). In schimb, se observă îmbunătățirea semnificativă a indicatorilor de performanță în cazul detecției inserțiilor / deleților cu ajutorul DeepVariant. De asemenea, se observă creșterea ușoară a specificității (*Recall*) fără o afectare semnificativă a specificității (*Precision*) daca se analizează varianta pre-procesata (GATK4) a acestui set de date (Fig. 2.6). Însă, aceste diferențe nu sunt suficient de semnificative pentru a justifica dublarea timpului de rulare al analizei bioinformatice. Timpul de rulare al analizei DeepVariant a fost de 22884.45 ± 34.3 secunde în cazul serverului HPC, respectiv 25304.84 ± 212.18 secunde in cazul VM (cloud). In timp ce durata de rulare a GATK4 *HaplotypeCaller*, a fost de 29750.28 secunde. Deși timpul de rulare al celor metode de analiza este comparabil, timpul total de execuție este redus prin intermediul accelerării mapării cu ajutorul GPU si prin eliminarea etapei de pre-procesare GATK - la 7.9h pentru setul de date SRR6794144 (NA12878 / HG001).

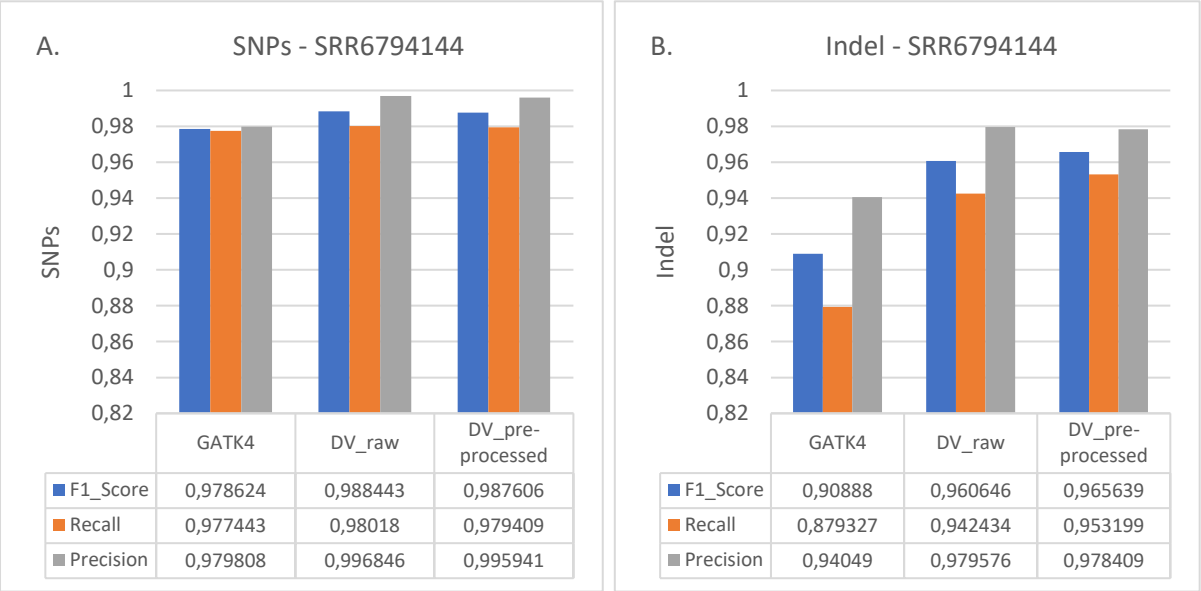


Figura 2.8 Indicatori de performanta pentru proba SRR6794144 (NA12878 / HG001) la nivel de SNP (A) si INDEL (B) analizata cu ajutorul versiunii GRCh38 a genomului de referință.

In cadrul proiectului a fost prevăzută si o etapa de evaluare a performanțelor fluxurilor de lucru pentru analiza datelor NGS cu seturi de date NGS clinice furnizate de către firma Genetic Lab SRL (C.U.I 13571513), Bucuresti, Romania: un genom (WGS) cu acoperire de 30x, secvențiat cu ajutorul platformei Illumina Hiseq si cinci exom-uri (WES) cu acoperire 100x, secvențiate cu ajutorul platformei Illumina NovaSeq 6000 si kitul de secventiere Agilent SureSelect Human All Exon V6. Datele WGS si WES brute au fost analizate cu fluxurile de lucru validate GATK 4.2.0 si DeepVariant, folosind genomul de referința uman GRCh38. Seturile de date WES au fost analizate doar cu fluxul de lucru DeepVariant rulat cu opțiunea WES. Variantele scurte (SNP si INDEL) identificate cu ajutorul celor doua fluxuri de lucru au fost comparate cu cele furnizate de către firma Genetic Lab (GL) împreună cu datele NGS brute. Fișierele VCF furnizate de către firma GL au fost produse in urma analizei independente a acelorași seturi de date NGS, cu ajutorul unui proceduri de lucru bazată tot pe pachetul GATK. In cazul setului de date WGS se observa o variație cuprinsa intre 3.9 - 8.5% a numărului de variante SNP si INDEL identificate de firma GL si cele identificate cu fluxurile de lucru dezvoltate in cadrul proiectului. In cazul celor cinci seturi de date WES, variația este cuprinsa, in general, intre 2 - 10%, dar cu un maxim de 14.8% pentru setul de date 220236_VA (Tabel 1). Variația observata, in special in cazul setului de date WGS, ar indica o scădere a sensibilității de detecție a variantelor SNP cu cel puțin 6% pentru ambele fluxuri de lucru. Din păcate, este foarte complicat de evaluat performantele

fluxurilor de lucru GATK si DeepVariant cu ajutorul unor seturi de date NGS nestandardizate, in principal pentru ca nu se poate determina sensibilitatea si nici specificitatea celor doua metode fără sa se cunoască variantele adevărate, dar si pentru faptul ca acești parametrii pot prezenta o variație neliniara in funcție de acoperirea mediana (număr median de citiri), complexitatea regiunilor (compoziția bazelor GC) si de nivelul de zgomot al citirilor (baze N).

Tabel 1 Comparația variantelor scurte SNP si INDEL obținute de catre Genetic Lab cu cele obținute cu fluxurile de lucru dezvoltate in cadrul proiectului

Sample	Method	Type	SNP variants	INDEL variants
LS221911	Genetic Lab	WGS	4111650	905539
LS221911	DeepVariant	WGS	3761002	869619
LS221911	GATK 4.2.0	WGS	3838770	848843
225282_MM	Genetic Lab	WES	330848	50952
225282_MM	DeepVariant	WES	361596	46847
225312_IC	Genetic Lab	WES	353373	52877
225312_IC	DeepVariant	WES	372431	47272
219493_II	Genetic Lab	WES	277453	44420
219493_II	DeepVariant	WES	311875	41439
220236_VA	Genetic Lab	WES	347812	54457
220236_VA	DeepVariant	WES	399484	52560
225269_MR	Genetic Lab	WES	366946	56653
225269_MR	DeepVariant	WES	389357	50290

Acesta comparație subliniază importanta validării fluxurilor de lucru pentru analiza datelor NGS. Având in vedere că validarea fluxului de lucru DeepVariant s-a realizat doar pentru WGS, mai departe s-a efectuat validarea acestui flux de lucru pentru seturile de date WES.

Validarea fluxului de lucru DeepVariant pentru analiza seturilor de date exomice (WES) a fost relizata cu ajutorul a șapte seturi de date GIAB: NA12878 (HG001), membrii trioului (familiar) Ashkenazi (HG002 - HG004) si membrii trioului Chinese Han (HG005 - HG007). Seturile de date brute WES au fost descărcate din baza de date NCBI Sequencing Read Archive (SRA), folosind codurile de acces (ERR1905890, SRR2962669, SRR2962692, SRR2962694, SRR2962693, SRR14724507, SRR14724506). Seturile de date HG001 - HG005 au fost obținute cu ajutorul kitului de secventiere Agilent SureSelect All Exon v5, iar seturile HG006 - HG007 cu Agilent SureSelect All Exon v7. Seturi utilizate in procedura de validare au un grad ridicat de acoperire medie (100-200x). Variantele adevărat pozitive (format BED) si pozițiile regiunilor genomice cu grad înalt de încredere au fost descărcate de pe serverul GIAB FTP. Pe scurt, următoarea procedură a fost aplicată pentru analiza celor șapte seturi de date WES - citirile brute NGS au fost aliniate pe genomul de referință GRCh37 cu programul BWA MEM, pre-procesarea cu GATK in vederea marcării citirilor duplicate, urmată de procesarea cu DeepVariant rulat cu protocolul optimizat pentru seturilor de date exomice. Determinarea sensibilității (recall) si specificității (precision) s-au determinat cu pachetul software hap.py. Regiunile genomice cu grad înalt de încredere pentru seturile WES individuale au fost intersectate cu ajutorul programului BEDtools, si s-au păstrat doar regiunile regăsite in toate seturile de date.

Rezultatele validării fluxului de lucru Deep Variant cu protocolul WES sunt in concordanta cu rezultatele publicate pentru seturile de date exomice HG001 - HG007 [21] (Tabel 2). Valorile de sensibilitate si specificitate înregistrate pentru seturile de date HG001 - HG005 au fost, in general, de >0.99, dar cu câteva abateri in cazul variantelor INDEL. Valorile ceva mai scăzute obținute in cazul seturilor de date HG006 - HG007 ar putea fi explicate prin faptul ca, numai 95% din regiunile cu grad înalt de încredere au avut o acoperire de cel puțin 10x, in ciuda gradului ridicat de acoperire medie (170 - 180x) [21].

Tabel 2 Indicatorii de performanta pentru validarea fluxului de lucru DeepVariant cu protocolul WES

Sample	Variant type	Recall	Precision	F1 Score
HG001	SNP	0.993206	0.998413	0.995803
HG001	INDEL	0.992191	0.997548	0.994862
HG002	SNP	0.99436	0.998878	0.996614
HG002	INDEL	0.983444	0.99618	0.989771
HG003	SNP	0.993699	0.998115	0.995902
HG003	INDEL	0.981077	0.992658	0.986834
HG004	SNP	0.994963	0.998994	0.996974
HG004	INDEL	0.989831	0.997079	0.993441
HG005	SNP	0.99653	0.999321	0.997924
HG005	INDEL	0.989389	0.995939	0.992653
HG006	SNP	0.98545	0.995834	0.990615
HG006	INDEL	0.957627	0.991266	0.974156
HG007	SNP	0.984973	0.995764	0.990339
HG007	INDEL	0.963675	0.99345	0.978336

2.3.2 Identificarea variantelor scurte din linie somatica (MuTect2 si Strelka2) si CNV

Identificarea variantelor somatice la un nivel ridicat de precizie si sensibilitate este foarte complicata, in principal datorită eterogenității tumorale (i.e. populații celulare subclonale), artefactelor de secventiere si aliniere, precum si contaminării cu celule normale [22]. Cancerul mamar este un termen generic care descrie o colecție eterogena de transformări neoplazice maligne. Probele tumorale sunt de cele mai multe ori eterogene la nivel morfologic, si cuprind diferite tipuri celulare în compoziție stromală variată. Diversitatea histologică și moleculară inter-tumorala a fost asociată diferitelor fenotipuri clinice care privesc potențialul de metastazare și răspunsul terapeutic. Mutațiile somatice stau la baza mecanismului de transformare a unei celule normale intr-una tumorală. Similar mutațiilor linei germinative, lungimea secvenței nucleotidice afectate poate varia de la o singura baza nucleotidica (SNP) la un cromozom întreg. Studiile genomice realizate cu sprijinul secventierii de noua generație au avut ca scop: identificarea genelor care determină progresia cancerului, clasificarea subtipurilor de cancer in vederea stabilirii unei corelații intre proprietățile moleculare și evoluția clinica, pentru determinare țintelor terapeutice, precum si asocierea factorilor de mediu de tiparele mutaționale. Multe din metodele de procesare si identificare a variantelor somatice generează rezultate discrepante

pentru un număr mare de alele. Prin intermediul inițiativei ICGC-TCGA DREAM Somatic Mutation Calling Challenge (SMC-DNA) s-a încercat stabilirea unui consens intre rezultatele diferitelor metode / protocoale de analiză a variantelor somatice [23]. S-au generat *in silico* 6 seturi de date NGS brute normale si tumorale: probele normale s-au obținut prin secventierea cu acoperire ridicată (citiri 60-80X) a unor linii celulare, iar cele tumorale au fost obținute prin inserția in probele normale a doua spectre mutaționale nesuprapuse i.e. unele selectate randomic si altele care au țintit gene asociate diferitelor tipuri de cancer. Aceste seturi de date au fost folosite pentru a valida fluxurile de lucru pentru identificarea variantelor somatice.

S-au folosit trei tipuri de seturi de date pentru validarea fluxurilor de lucru optimizate pentru detecția variantelor somatica din genomuri / exomuri umane: (1) primele 4 seturi de date ICGC-TCGA DREAM Somatic Mutation Calling Challenge (2) Mixtura seturilor de date GIAB (Genome in a Bottle) NA12878 si NA24385 realizat la Centrul Medical Hartwig din Amsterdam si (3) un set de date realizat *in silico* cu un anumit număr de SNP si INDEL introduse in proba „tumorala" generata *in silico*. Pe lângă aceste seturi de date, s-a mai generat încă un set de date pentru validarea fluxului de lucru CNV.

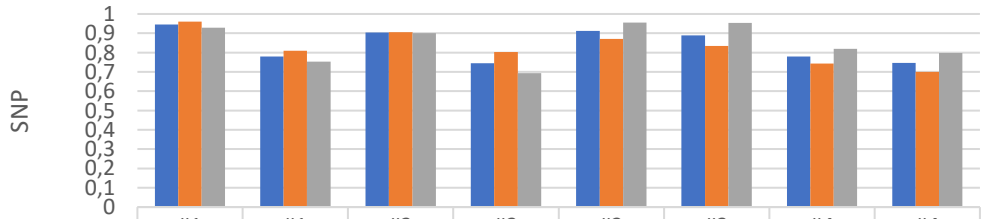
Probele normale ICGC-TCGA DREAM au obținute prin secventierea cu acoperire ridicată (citiri 60-80X) a unor linii celulare, iar cele tumorale au fost obținute prin inserția in probele normale a doua spectre mutaționale nesuprapuse i.e. unele selectate randomic si altele care au țintit gene asociate diferitelor tipuri de cancer (mamar si de prostată). Primele trei seturi sintetice sunt de sex feminin si prezintă gene derivate din linia celulara HCC1143 BL (linie celulara stabilizata de carcinom mamar). Primele doua seturi nu prezinta populații subclonale i.e. o singura clona si nu prezinta INDEL, ci doar SV (deleții, inserții, translocații si inversii). Setul 3 prezintă frecvențe alelice subclonale de 50, 33 si 20%. Setul 4 este singurul de sex masculin, si prezintă frecvențe alelice subclonale de 30 si 15%. Acest set conține gene asociate cancerului de prostata. Probe tumorale sintetice 3 si 4 prezintă atât variante somatice scurte, dar si SV. Setul de date mixt NA12878/NA24385, asemănător unei tumori, a fost realizata prin combinarea celor doua genomuri GIAB pentru validarea metodelor de identificare a mutațiilor somatice cu frecventa scăzută. In acest context, variantele scurte SNP/INDEL specifice genomului NA12878 sunt identificate drept modificări genetice "somatice", iar variantele comune NA12878/NA24385 sunt „germinative". Proba tumorala WGS (citiri pereche de 150 bp) cu acoperire de 90x, a fost generata prin amestecarea ADN NA12878 in proporție de 30% cu ADN provenit de la proba NA24385 in proporție de 70%. Frecventele alelice somatice fiind astfel de 15% pentru heterozigoți, respectiv 30% homozigoți din NA12878. Proba normala a acestui set de date a fost obținută prin secventierea WGS (citiri pereche de 150 bp) a NA24385 la o acoperire de 30x. In ultima etapă a procesului de validarea a fluxurilor de lucru, s-a utilizat programul VarSim [24] pentru a genera noi seturi de date sintetice. Seturile de date, generate complet *in silico* se pot utiliza pentru a controla cu exactitate frecvența alelică (i.e. simularea diferitelor grade de contaminare cu celule normale), dar și a numărului de variante ce se pot introduce in proba tumorală. Pentru generarea seturilor de date s-a folosit versiunea UCSC GRCh37/hg19 a genomului de referință uman. Ambele seturi de date au fost construite cu următorii parametrii comuni: lungimea citirilor perechi de 100 bp, lungimea medie a fragmentelor de 350 bp cu deviație standard 50 bp, acoperire de 50x, precum si utilizarea a 3 linii de secventiere cu acoperire egală. Pentru setul tumoral s-au utilizat mutațiile codante si necodante din baza de date COSMIC [25] pentru a insera la nivel genomic un număr de 6.000 SNP, 2.000 inserții si 2.000 deleții (4.000 INDEL). Prin amestecarea unei linii de secventiere normală cu 2 linii tumorale,s-a obținut o frecventa alelică homozigotică de aproximativ 60% si heterozigotica de 30% pentru variantele somatice, iar prin amestecarea a 2 linii normale cu o line tumorala s-au obținut frecvente alelice homozigotice de 30% si heterozigotice de 15%. Aceste seturi de date au fost folosite in procesul de validare a fluxurilor de data Mutect2 si Strelka2 pentru a evalua performantele celor două metode in cazul contaminării ușoare, respectiv contaminare severă. In scopul validării fluxului de lucru pentru identificarea si vizualizarea CNV, s-a utilizat programul CNV-Sim [26] pentru generarea *in silico* a probelor normale si tumorale. Si de aceasta data s-a utilizat versiunea UCSC GRCh37/hg19 a genomului de referință uman pentru ambele probe, dar modificările genomice au fost aplicate pe fiecare cromozom in parte (1 - Y). Astfel, s-au generat 24 de fișiere FASTQ per probă, contând citiri Illumina perechi, cu lungimea de 100 bp, si

acoperire de 30x. Setul de date tumoral a fost modificat in mod aleatoriu cu 20 CNV per cromozom, având o lungime cuprinsă intre 0.5 Mb si 10 Mb. Inserțiile si delețiile introduse la nivel genomic, in raport de 1:1, au crescut sau au redus diferitele segmente cromozomiale în intervalul 2-4.

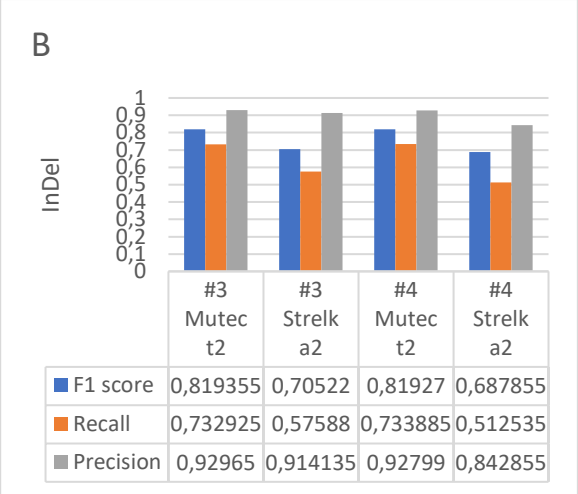
Seturile de date DREAM Nr. 1-3 au fost descărcate din baza de date NCBI SRA (coduri de acces: SRX570726, SRX1025978, respectiv SRX1026041) si s-a utilizat SRA Toolkit pentru obținerea perechilor de fișiere FASTQ nealiniate. Seturile de date DREAM Nr. 4 și NA12878/NA24385 au fost descărcate dintr-o alta sursa [27] în format BAM. Primele trei seturi de date DTREAM, precum si seturile de date generate *in silico* au fost supuse întregii etape de pre-analiza, iar cele in format BAM au pornit din etapa de indexare si sortare. Premergător alinierii, s-a utilizat programul Trimmomatic [28] pentru tăierea capetelor (5' - 3') citirilor si eliminarea adaptorilor de secventiere NGS. Fișierele FASTQ procesate au fost aliniate fie pe versiunea GRCH38, fie pe versiunea GRCh37/hg19 a genomului de referință uman, cu ajutorul programului BWA-MEM v0.7.17-1. Citirile aliniate au fost convertite in fișiere BAM, si mai departe au fost sortate si indexate cu ajutorul programului SAMtools [7] iar duplicatele au fost marcate. Identificarea efectivă a variantelor a fost realizata cu ajutorul modulului GATK MuTect2. Variantele rezultate au fost mai departe filtrate cu ajutorul modulului GATK FilterMutectCalls pentru excluderea fals pozitivelor. In final, variantele au fost adnotate cu ajutorul modulului GATK Funcotator. Fluxul de lucru bazat pe programul Strelka2 a fost utilizat în același modalitate - analiza in tandem a probelor tumorale si normale în vederea evidențierii transformărilor neoplazice la nivel genomic. Pentru detecția variantelor structurale (inversii, translocări, inserții si deleții > 1 Kb - 5 Mb) s-a utilizat programul Manta [29], dar nu a fost integrat intr-un flux de lucru. Variantele somatice identificate au fost filtrate adițional, in vederea îmbunătățirii parametrilor de performanta, cu ajutorul programului BCFtools [8]. Evaluarea parametrilor de sensibilitate (Recall), specificitate (Precision) si scorul F1 s-a calculat cu ajutorul programului RTG-Tools v3.8.4 [30] cu metoda de analiză vcfeval.

Cele două fluxuri de lucru au identificat in proporție de 98 - 99% SNP din cele patru seturi de date DREAM, dar sensibilitatea detecției INDEL (seturile DREAM Nr. 3-4) a fost de maxim 87% la evaluarea variantelor candidate (sensibilitate maxima si specificitate minimă). Principala provocare in detecția variantelor scurte SNP și INDEL somatice fiind diferențierea variantelor adevărat pozitive (TP) de fals pozitive (FP) - metodele de filtrare, folosite de ambele programe, au favorizat reducerea minimă a sensibilității in defavoarea specificității. Pentru ambele seturi de rezultate DREAM Nr. 1-4, s-a efectuat o prima optimizare, cu scopul îmbunătățirii parametrului F1, dar cu posibilitatea ajustării individuale a sensibilității si specificității, in funcție de aplicație. Optimizarea rezultatelor DREAM nu a necesitat decât filtrarea suplimentară a variantelor, folosind 1-2 parametrii de calitate disponibili in fișierul VCF.

A



	#1 Mutect2	#1 Strelka2	#2 Mutect2	#2 Strelka2	#3 Mutect2	#3 Strelka2	#4 Mutect2	#4 Strelka2
F1 score	0,9448	0,77988	0,90354	0,74456	0,91167	0,88986	0,77967	0,74638
Recall	0,9607	0,80887	0,90604	0,80309	0,87118	0,83373	0,74305	0,70049
Precision	0,92943	0,75289	0,90105	0,69399	0,95611	0,95409	0,8201	0,79869



	#3 Mutect2	#3 Strelka2	#4 Mutect2	#4 Strelka2
F1 score	0,819355	0,70522	0,81927	0,687855
Recall	0,732925	0,57588	0,733885	0,512535
Precision	0,92965	0,914135	0,92799	0,842855

Figura 2.9 Indicatori de performanta pentru detecția SNP din seturile de date DREAM Nr. 1-4 (A) si INDEL pentru seturile de date DREAM Nr. 3-4 (B).

In cazul primului set sintetic DREAM, s-a îmbunătățit rezultatele Mutect2 (SNP) prin creșterea specificității cu 7,5%, păstrând valoarea sensibilitatea neschimbată (96%). Rezultatele Strelka2 (SNP) pentru același set de date au fost mult mai slabe pentru specificitate si sensibilitate.

In cazul rezultatelor de validare Mutect2 obținute cu setul de date DREAM Nr. 2, s-a îmbunătățit specificitatea cu 22%, cu prețul reducerii sensibilității cu doar 4%, iar scorul F1 a fost îmbunătățit cu 12%.

Indicatorii de performantă pentru seturile de date sintetice DREAM Nr. 3 si 4, pentru ambele fluxuri de lucru, fiind mult mai apropiate. Si de data acesta fiind necesară optimizarea specificității metodelor. Astfel, se înregistrează valori bune ale detecției SNP, in special pentru setul Nr. 3 - acesta fiind cel mai reprezentativ pentru o proba tumorala reală, datorita prezenței a 3 populații subclonale de variante somatice. In schimb, s-au înregistrat valori ceva mai slabe de sensibilitate pentru detecția INDEL cu fluxul de lucru Strelka2 (Fig. 2.9B).

Per total, fluxul de lucru Mutect2 s-a dovedit a fii superior în ceea ce privește detecția variantelor somatice scurte SNP si INDEL.

Parametrii utilizați pentru filtrarea post-analiză a variantelor somatice din seturile de date DREAM cu ambele fluxuri de lucru nu au fost adecvați pentru filtrarea seturilor de date GIAB (NA12878/NA24385) - fiind necesară filtrarea de la zero a variantelor candidate, individual pentru SNP si INDEL, folosind pana la 4 parametrii de calitate pentru Mutect2 si pana la 3 parametrii pentru Strelka2.

Deoarece variantele "somatice" prezente în mixtura NA12878/NA24385 sunt, de fapt, variante germinative, a fost necesară rularea fluxului de lucru Mutect2 fără PON (Panel de Normali).

In procesul de validarea al fluxului de date Mutect2 cu acest set de date, s-a obținut cele mai ample diferențe dintre valorile standard (sensibilitate si specificitate) si cele optimizate. De această data, pentru detecția SNP s-a îmbunătățit atât sensibilitatea (29,7%) cât si specificitatea (4,9%). Pentru detecția INDEL, a rezultat o creștere a sensibilității cu 55% si a specificității cu 3%. Rezultatele obținute cu fluxul de lucru Strelka2 au înregistrat o creștere de numai câteva puncte procentuale. În final s-au obținut rezultate foarte apropiate prin cele doua fluxuri de lucru (Fig. 2.10).

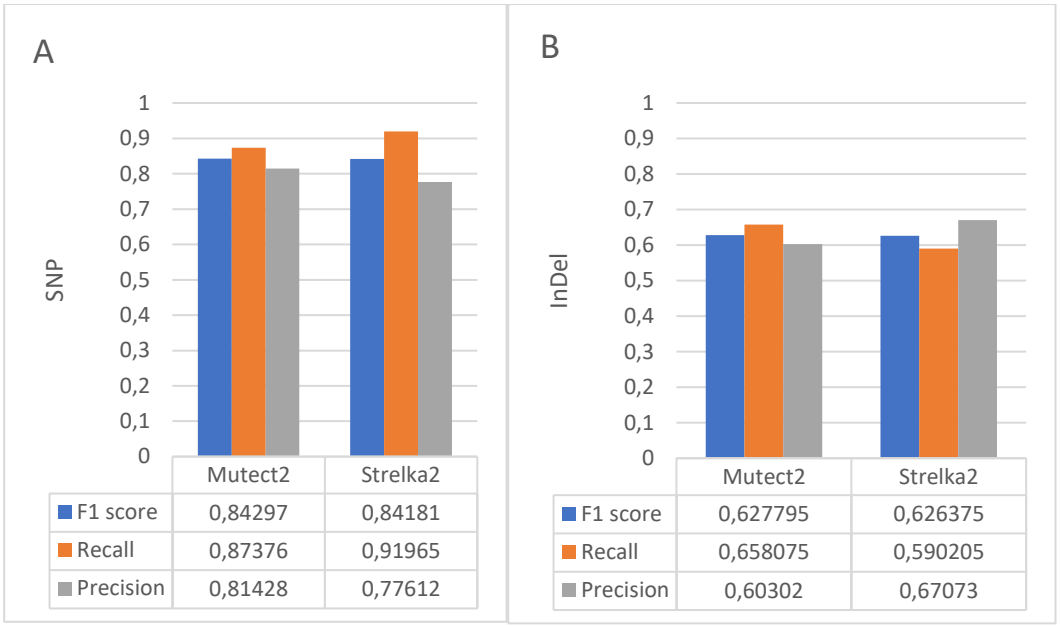
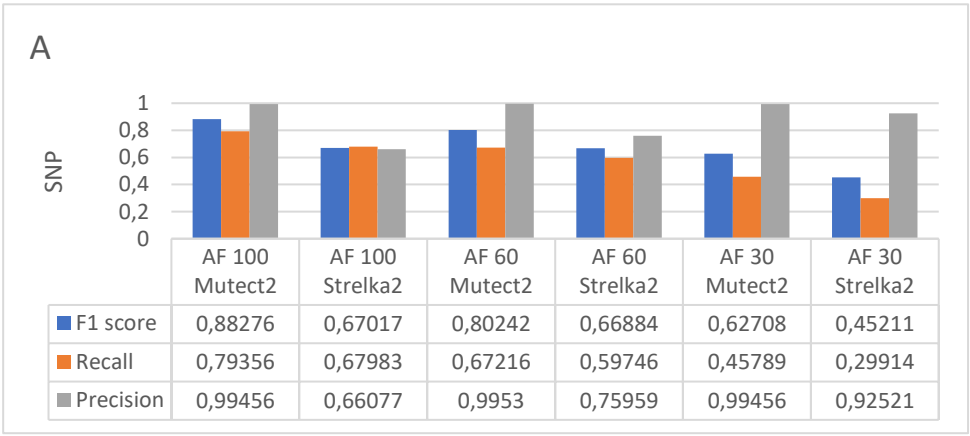
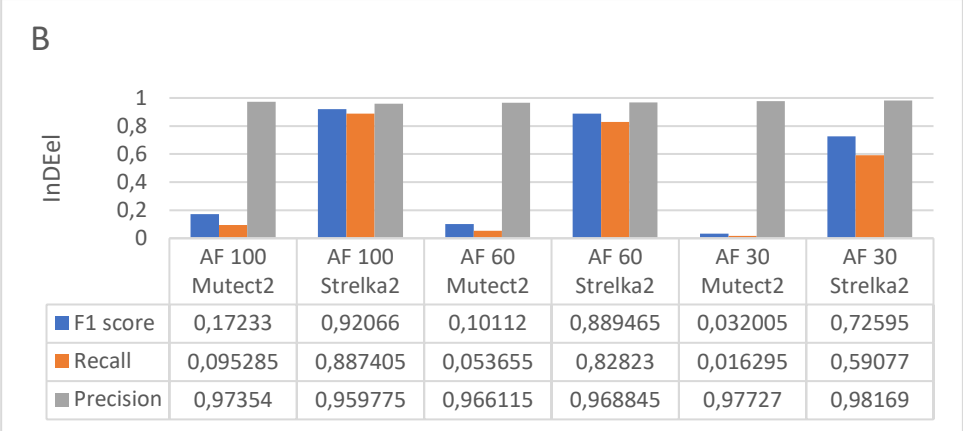


Figura 2.10 Indicatori de performanta pentru detecția SNP (A) si INDEL (B) din seturile de date NA12878-NA24385

In final, utilizarea seturilor de date generate *in silico* cu ajutorul programului VarSim au relevat diferențe interesante intre cele două metode de analiză.

În cazul setului de date standard - frecvență alelică nemodificată "AF 100" - s-au obținut rezultate rezonabile cu ajutorul fluxului de lucru Mutect2: sensibilitate de 79,3% si specificitate foarte bună de 99,4%. Reducerea frecvențelor alelice cu 1/3 "AF 60" a condus, in mod așteptat, la reducerea sensibilității la 67.2%, iar în cazul reducerii frecventelor alelice cu 2/3 "AF 30" s-a redus sensibilitatea la 29.9%. În toate cele trei situații, specificitatea s-a menținut peste 99%. Din păcate, și cu aceste seturi de date, rezultatele obținute cu fluxul de lucru Strelka2 au fost suboptime. În mod contraintuitiv, specificitatea metodei a crescut invers proporțional cu frecvența alelică (Fig. 2.11A). Singura situație in care fluxul de lucru Strelka2 a prezentat un avantaj, prin comparație cu fluxul de lucru Mutect2, a fost în identificarea INDEL prezente în aceste seturi de date. Cauza acestei discrepanțe nefiind clara - din setul de date "AF 100" s-au identificat, prin metoda MuTect2, maxim 9,5% INDEL, iar prin metoda Strelka2 s-au identificat maxim 88,7%. Si de această dată fiind notabilă creșterea specificității invers proporțional cu frecvența alelică in cazul fluxului de lucru Strelka2 (Fig. 2.11B).





	AF 100 Mutect2	AF 100 Strelka2	AF 60 Mutect2	AF 60 Strelka2	AF 30 Mutect2	AF 30 Strelka2
F1 score	0,17233	0,92066	0,10112	0,889465	0,032005	0,72595
Recall	0,095285	0,887405	0,053655	0,82823	0,016295	0,59077
Precision	0,97354	0,959775	0,966115	0,968845	0,97727	0,98169

Figura 2.11 Indicatori de performanta pentru detecția SNP (A) si INDEL (B) din seturile de date generate *in silico* AF 100-30.

Variantele structurale sunt definite drept modificări genetice de > 1 Kb, dar mai mici de 5 Mb, si pot include inversii si translocații echilibrate (diploide) sau pot introduce dezechilibre genomice (inserții si deleții), denumite si CNV.

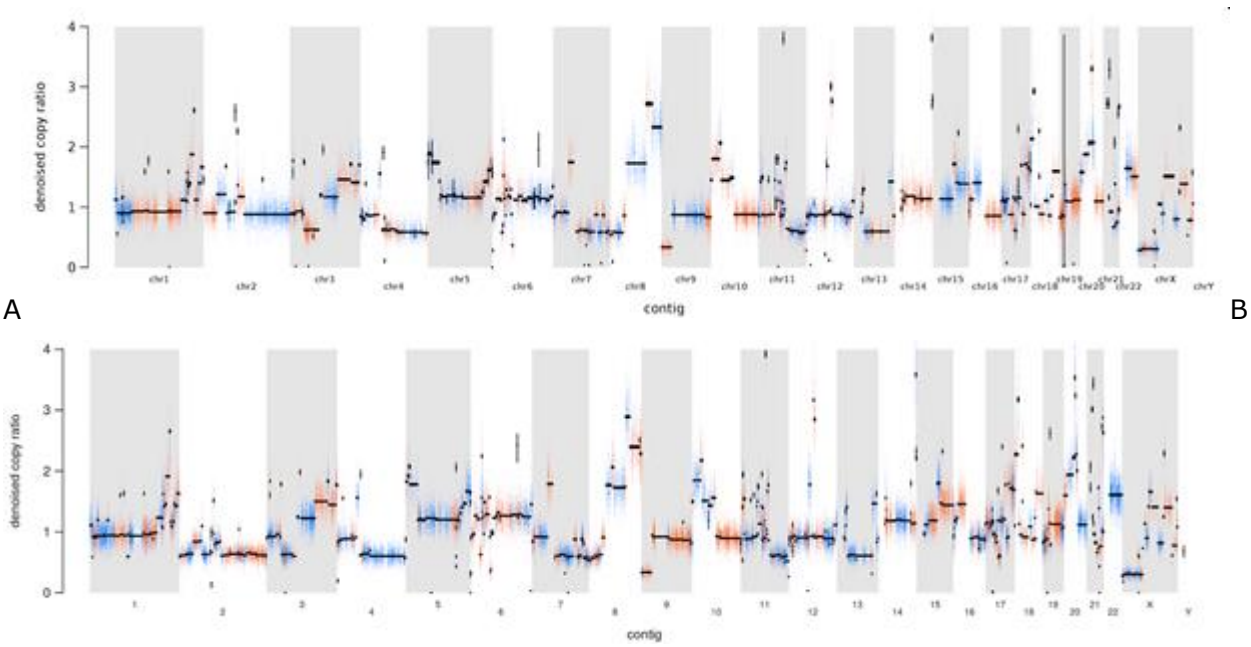


Figura 2.12. Vizualizare CNV identificate in proba tumorală HCC1143 folosind versiunea GRCh38 (A) a genomului de referință uman si versiunea UCSC GRCh37/hg19 (B).

Prima etapă a procesului de validare a fluxului de lucru destinat identificării CNV a fost testarea setului de date obținut prin secventierea linei celulare HCC1143 (proba tumorală) si a linei complementare HCC1143 BL (probă normală), utilizând versiunea USCS GRCh37/hg19 a genomului de referință uman. Rezultatele au fost comparate cu cele obținute cu versiunea GRCh38 (Fig. 2.12). A doua etapă a fost validarea fluxului de lucru CNV cu ajutorul a două seturi de date generate *in silico*. Pentru testarea rezoluției de detecție a metodei a fost construit un set secundar de date prin amestecarea contig-urilor cromozomiale normale printre cele tumorale (Fig. 2.13B). Rezultatele indica cât se poate de clar segmentele cu număr de copii modificate (inserții si deleții), si corespund pozițiilor CNV introduse de programul CNV-Sim. (Fig. 2.13).

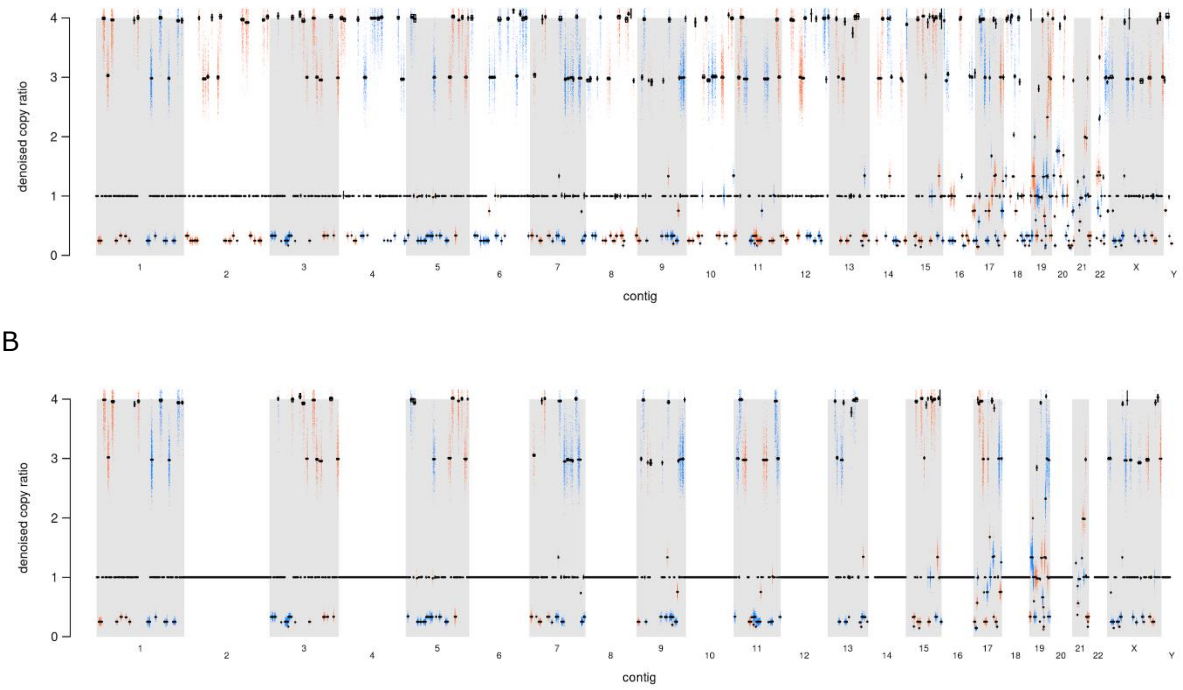


Figura 2.13 Vizualizare CNV identificate in proba tumorală generată *in silico* cu ajutorul programului CNV-Sim (A), respectiv pentru o probă cu contig-uri normale inserate printre cele tumorale (B).

Deși CNV reprezintă un subset al SV, fluxul de lucru GATK realizat pentru identificarea CNV nu poate diferenția inversiile si translocările echilibrate i.e. celelalte tipuri de variantele structurale. Primele teste ale fluxului de lucru realizat in vederea identificării tuturor variantelor structurale (Manta) au indicat o sensibilitate de aproximativ 60% pentru seturile de date DREAM Nr. 1-3.

2.4 Utilizarea GPU

S-au implementat diferite masuri necesare pentru reducerea timpului de procesare al fluxurilor de lucru GATK4 si DeepVarinat. S-a adăugat posibilitatea de a selecta, pe lângă programul BWA MEM, unul din cele doua programe ce apelează NVIDIA CUDA (Compute Unified Device Architecture) pentru paralelizarea alinierii citirilor NGS la genomurile de referință cu ajutorul Nvidia GPGPU.

Programul SOAP (*Short Oligonucleotide Analysis Package*) a fost printre primele programe de aliniere a datelor NGS, iar versiunea modernă a acestuia (SOAP3-dp) [31] aduce o serie de optimizări precum utilizarea indexării comprimate si a programării dinamice de optimizare a memoriei VRAM. Pe scurt, procesarea datelor SOAP3-dp se realizează in trei etape: 1) se utilizează algoritmul BWT bidirecțional accelerat cu ajutorul GPU pentru alinierea citirilor fără a introduce goluri la nivelul secvenței de referință; 2) utilizarea algoritmului de programare dinamică accelerat pe GPU pentru alinierea citirilor rămase nealiniate cu ajutorul citirilor omologe deja aliniate; 3) se folosesc elemente extinse ale subcontig-urilor in încercarea de alinia citirile ramase nealiniate.

GASAL2 (*GPU Accelerated Sequence Alignment Library* v2) [32] este o bibliotecă GPU specializată pe alinierea locală, globală si semi-globală a secvențelor ADN si ARN. Funcțiile de aliniere GASAL2 sunt asincrone/neblocante și permit suprapunerea completă a execuției CPU și GPU. Alinierea globală a secvențelor este realiză prin algoritmul Needleman-Wunsch (NW), iar alinierea locala prin algoritmul Smith-Waterman (SW). Biblioteca GASAL2 conține funcții ce se pot integra în alte programe de analiză NGS precum gase-gasal2. Acest program utilizează fie

GASE (*Generic Aligner for Seed-and-Extend*) - o extensie a BWA (ver. 0.7.13) - sau BWA MEM pentru partea de indexare a citirilor si GASAL2 pentru accelerarea alinierilor.

Programul BarraCUDA utilizează algoritmul *Burrows-Wheeler Transform* (BWT) [33]. Algoritmului BWT procesează datele într-un ritm constant, iar timpul de rulare depinde, in principal, de numărul de perechi baze (Gbp) si de complexitatea datelor i.e. proporția de citiri eronate. Acest program a fost obținut prin rescrierea modulului de bază (BWT) pentru exploatarea paralelismului masiv al NVIDIA GPGPU (CUDA).

Comparația între timpului de rulare al alinierii citirilor NGS cu ajutorul programului BWA-MEM si timpii de rulare obținuți cu trei programe ce permit accelerarea mapării cu ajutorul GPU este prezentată in Tabelul 3. Din păcate, programul BarraCUDA rulează sub-optim pe acceleratoarele grafice. Timpul de rulare al setului de date SRR949537 este de 4 ori mai scurt decât BWA MEM rulat pe un singur nucleu CPU, dar de peste 2 ori mai extins daca se rulează maparea pe 24 nuclee CPU. Rulare sub-optimă a BarraCUDA se observă si din timpul de rulare al setului de date SRR099988, motiv pentru care nu s-a rulat maparea celorlalte seturi de date mult mai extinse. Cea mai semnificativă reducere a timpului de rulare s-a obținut cu ajutorul programului gase-gasal2 pentru seturile de date nu un număr redus de perechi de baze (SRR949537), respectiv un număr moderat de perechi de baze (SRR099988). Din cauza unei probleme de alocare a memoriei necesare, nu s-a putut mapa setul de date SRR6794144 și nici alte seturi, cu un număr mai mare de perechi baze, cu acest program. În schimb s-a putut accelera alinierea setului de date SRR17246885, ducând la înjumătățirea timpului de rulare. În final, programul SOAP3-dp nu a prezentat probleme tehnice în maparea celor 4 seturi de date, si permite reducerea timpul de rulare de 2.9 ori pentru setul SRR6794144, respectiv 3.6 ori pentru setul SRR17246885, prin comparație cu BWA-MEM (Tabel 1).

Tabel 3 Comparația între timpului de rulare al mapării citirilor NGS cu ajutorul programului BWA-MEM si timpii de rulare obținuți cu trei programe ce permit accelerarea mapării cu ajutorul GPU (secunde).

	BWA-MEM	**SOAP3-dp**		**gase-gasal2**		**BarraCUDA**	
NGS datasets	HPC	HPC	VM	HPC	VM	HPC	VM
SRR949537	178.01 ± 1.55	126.50 ± 0.46	127.57 ± 1.415	98.77 ± 0.47	111.85 ± 7.79	337.76 ± 1.82	335.71 ± 1.25
SRR099988	986.23 ± 5.66	686.31 ± 4.83	683.38 ± 7.52	543.14 ± 8.42	707.02 ± 5.29	2637.73 ± 27.61	2966.18 ± 5.56
SRR6794144	16897.892 ± 64.107	5768.25 ± 6.73	5672.36 ± 2.78	n/a	n/a	n/a	n/a
SRR17246885	20641.92 ± 5.349	5647.51 ± 4.33	5620.33 ± 12.27	10965.47 ± 40.37	11003.55 ± 18.69	n/a	n/a

3. Integrarea fluxurilor de lucru in serviciile de analiza NGS

3.1 Optimizarea fluxurilor de lucru

La finalul activitatii de dezvoltare s-au realizat o serie de modificări ale fluxurilor de lucru destinate analizei liniilor genetice germinative si somatice umane in vederea integrării acestora in serviciile de analiza a datelor de secvențiere de nouă generație (NGS) care urmează sa fie puse la dispoziția utilizatorilor in Centrul de Resurse Cloud și Big Data.

Fluxurile de lucru dezvoltate au fost actualizate pentru a permite descărcarea automata a datelor de secventiere obținute cu ajutorul platformei de secventiere Illumina din baza de date *European Nucleotide Archive*, folosind codul de acces.

Astfel, s-a adăugat posibilitatea ca toate fișierele FASTQ încărcate ca date de intrare sa fie recunoscute automat in etapa de post-procesare.

O alta îmbunătățire este posibilitatea de a selecta una din cele trei metode de mapare a citirilor NGS la genomurile de referință umane i.e. *Burrows-Wheeler* Aligner (BWA), *Short Oligonucleotide Analysis Package* (SOAP3-DP), si *GPU Accelerated Sequence Alignment Library* v2 (GASAL2).

Pentru testarea finala a fluxurilor de lucru ce analizează linia genetica germinativa s-a utilizat setul de date SRR099988 (11.9 Gbp), obținut cu ajutorul platformei Illumina HiSeq 2000. Testarea finala a fluxurilor de lucru pentru analiza liniei genetice somatice a fost realizata cu seturi de date genomice umane ICGC-TCGA DREAM Somatic Mutation Calling Challenge [23] Nr. 2 - proba normala SRR2020596 si proba tumorala SRR2020634, ambele de 56.5 Gbp, obținute tot cu ajutorul platformei Illumina HiSeq 2000.

Structura finala a celor 5 fluxuri de lucru dezvoltate care au fost integrate in platforma, precum si sursele datelor de intrare sunt prezentate mai jos.

3.2 Identificarea variantelor scurte SNP si INDEL cu DeepVariant

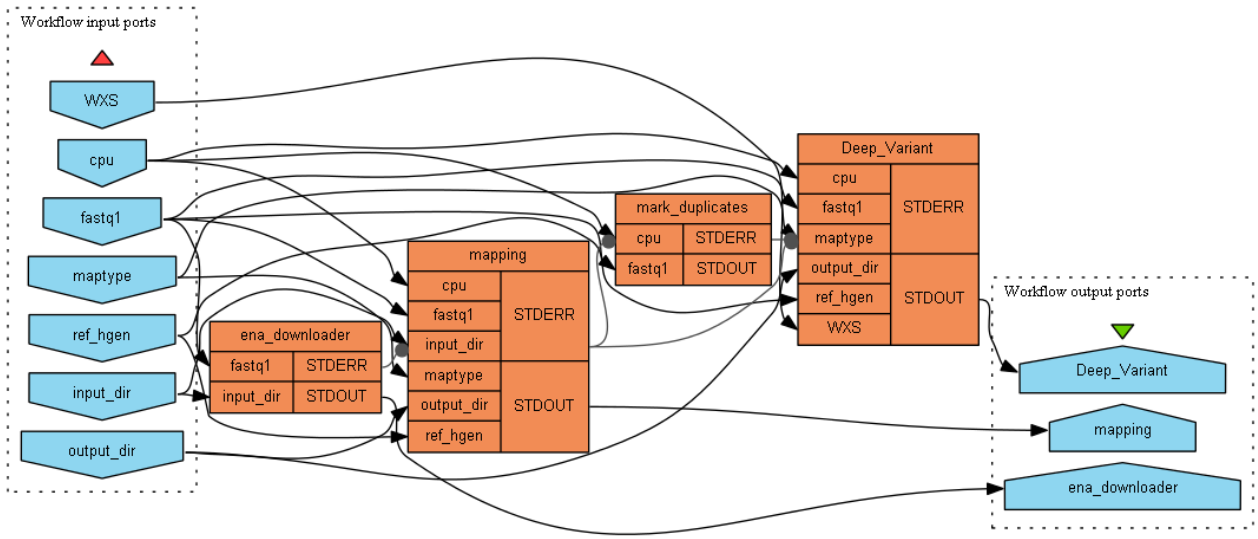


Figura 3.1 *Identificarea variantelor scurte SNP si InDel din linie genetica germinativa umana, utilizand DeepVariant.*

Date de intrare:

-inputvalue cpu: WGS / WES

-inputvalue cpu: e.g. 24

-inputvalue fastq1: e.g. SRR099988

-inputvalue input_dir: e.g. SRR099988

-inputvalue output_dir: e.g. SRR099988

-inputvalue maptype: bwa / soap3-dp / gasal2

-inputvalue ref_hgen: Homo_sapiens_assembly19 / Homo_sapiens_assembly38

3.3 Identificarea variantelor scurte SNP si INDEL cu GATK

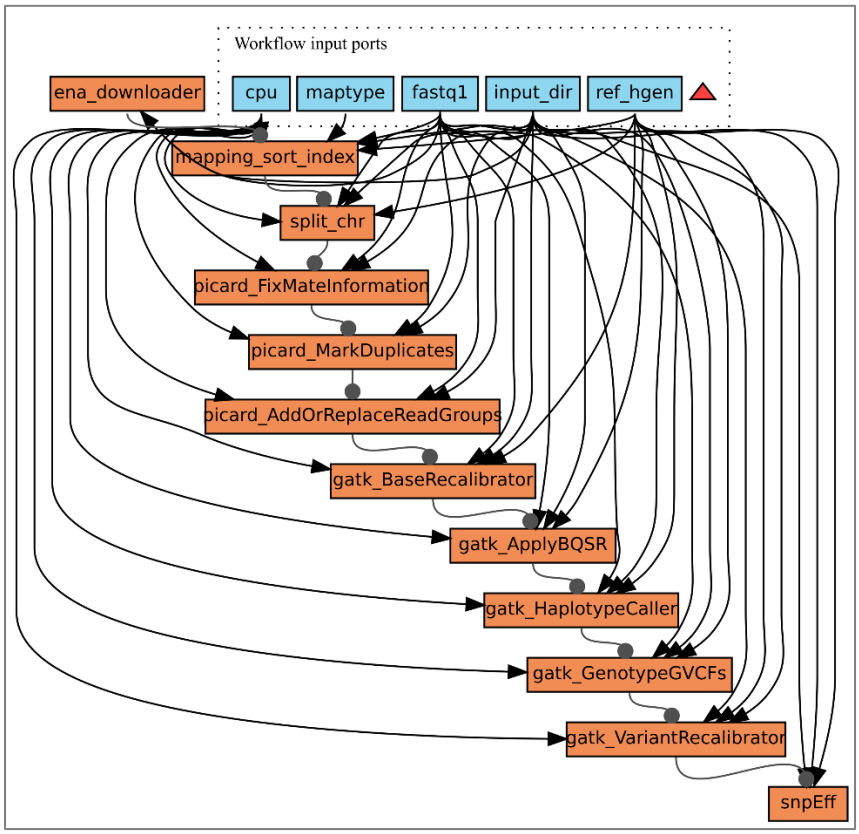


Figura 3.2 *Identificarea variantelor scurte SNP si InDel, din linie genetica germinativa umana, utilizand GATK.*

Date de intrare:

-inputvalue fastq1: e.g. SRR099988

-inputvalue input_dir: e.g. SRR099988

-inputvalue maptype: bwa / soap3-dp / gasal2

-inputvalue cpu: e.g. 24

-inputvalue ref_hgen: Homo_sapiens_assembly19 / Homo_sapiens_assembly38

3.4 Identificarea variantelor somatice umane cu MuTect2

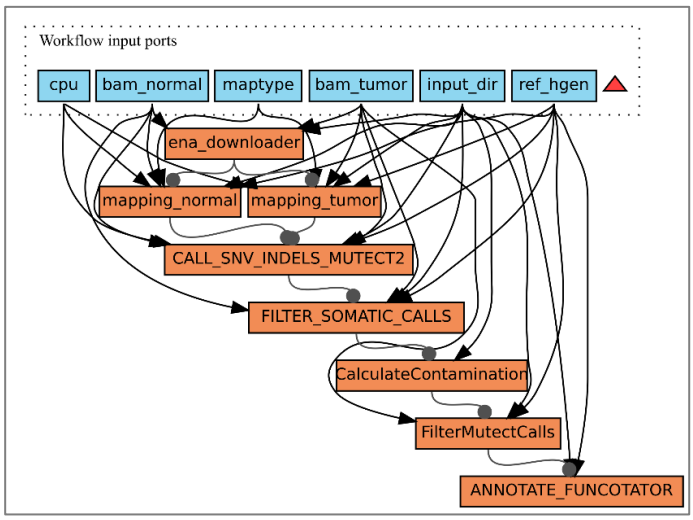


Figura 3.3 *Identificarea variantelor somatice umane din probe normale si tumorale, utilizand Mutect2 (GATK4).*

Date de intrare:

-inputvalue bam_normal: e.g. SRR2020596

-inputvalue bam_tumor: e.g. SRR2020634

-inputvalue input_dir: e.g. DREAM_S2

-inputvalue maptype: bwa / soap3-dp / gasal2

-inputvalue cpu: e.g. 28

-inputvalue ref_hgen: Homo_sapiens_assembly19 / Homo_sapiens_assembly38

3.5 Identificarea variantelor somatice umane cu Strelka2

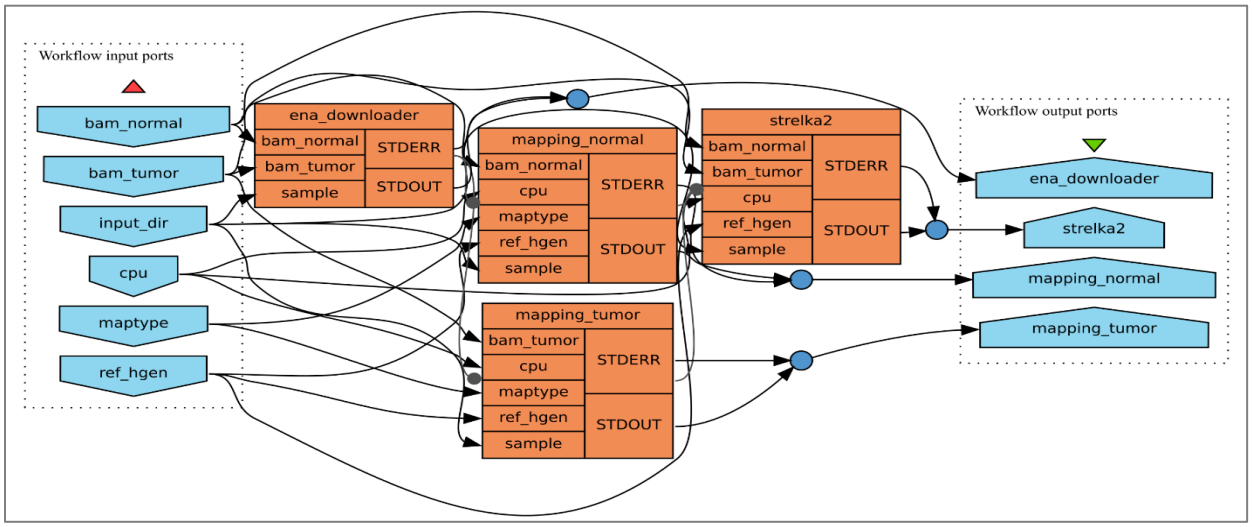


Figura 3.4 *Identificarea variantelor somatice umane din probe normale si tumorale, utilizand Strelka2.*

Date de intrare:

-inputvalue bam_normal: e.g. SRR2020596

-inputvalue bam_tumor: e.g. SRR2020634

-inputvalue input_dir: e.g. DREAM_S2

-inputvalue maptype : bwa / soap3-dp / gasal2

-inputvalue cpu: e.g. 10

-inputvalue ref_hgen: Homo_sapiens_assembly19 / Homo_sapiens_assembly38

3.6 Identificarea modificărilor genomice structurale - variante structurale si CNV

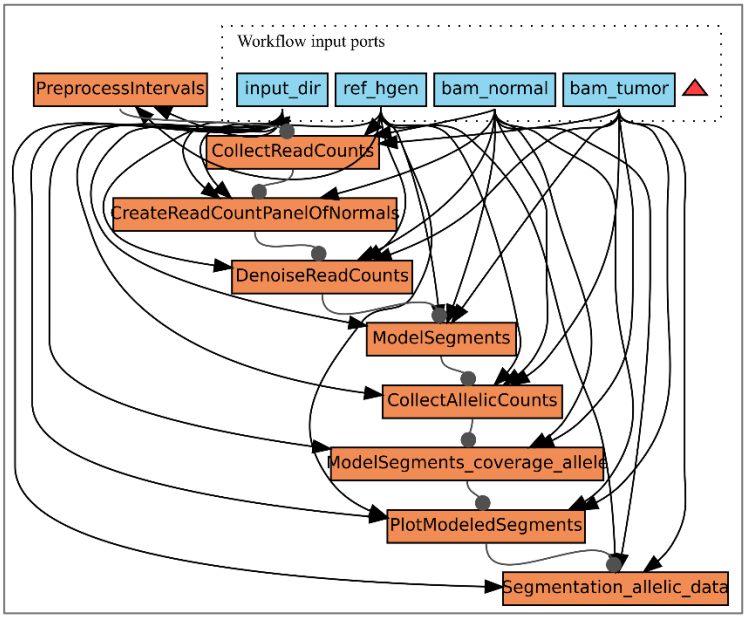


Figura 3.5 *Identificarea modificărilor genomice CNV din probele normale si tumorale, utilizand GATK.*

Date de intrare:

-inputvalue bam_normal: e.g. SRR2020596

-inputvalue bam_tumor: e.g. SRR2020634

-inputvalue input_dir: e.g. DREAM_S2_CNV

-inputvalue ref_hgen: Homo_sapiens_assembly19 / Homo_sapiens_assembly38

4. Platforma de analiza a datelor NGS

4.1 Procedura de initializare a fluxurilor de lucru NGS

Fisierele tip workflow (t2flow) necesare analizei datelor NGS pe platforma CCBD pot fi obtinute prin doua metode:

1) Utilizatorul descarca fisierele respective de pe platforme externe (cum este, de exemplu https://www.myexperiment.org);
2) utilizatorul generează fisierele t2flow pe calculatorul personal.

În cel de-al doilea caz, utilizatorul va folosi mai intai programul software *Taverna Workbench* instalat pe calculatorul personal pentru a construi grafic fisierul t2flow, pe care il salveaza local.

Etapa următoare constă în procesarea fluxului de lucru respectiv pe resursele hardware din centrul cloud.

Pentru aceasta, utilizatorul înregistrat va accesa mai întâi platforma software a CCBD, identificandu-se prin user și parola pe interfata descrisa in livrabilul 2.1.

Din interfata Biroului Virtual va apasa pe butonul „Applications" si apoi va selecta „TAVERNA" din bara cu aplicatiile pe care este autorizat sa le foloseasca.

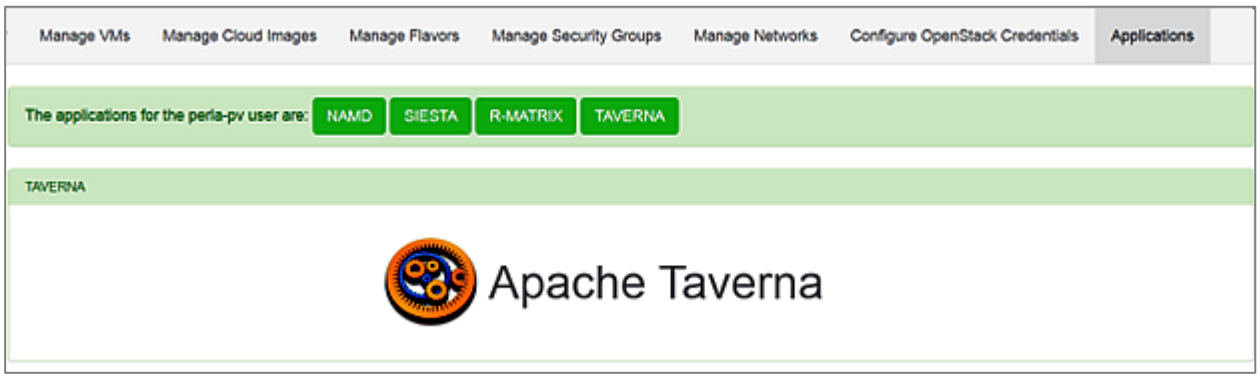


***Figura 4.1**: Antetul interfetei grafice de acces la aplicatia Taverna*

Pentru incarcarea in platforma a fisierului t2flow se foloseste butonul „Upload Workflow" din bara de comenzi a aplicatiei reprezentata in figura mai jos.

***Figura 4.2**: Bara de comenzi din interfata aplicatiei Taverna*

Actionarea butonului respectiv activeaza metoda grafica care a fost dezvoltata si implementata pentru incarcarea fisierelor de tip t2flow.

Se deschide fereastra din Fig. 4.3, care-i ofera utilizatorului posibilitatea sa incarce fisierul prin tragere si plasare (drag and drop) in fereastra de upload.

Daca sunt necesare mai multe fisiere t2flow, acestea se pot incarca simultan, intr-o singura operatiune, printr-o selectare multipla. Alternativ, fisierele pot fi trase cu mouse-ul din locatia lor in fereastra de upload.

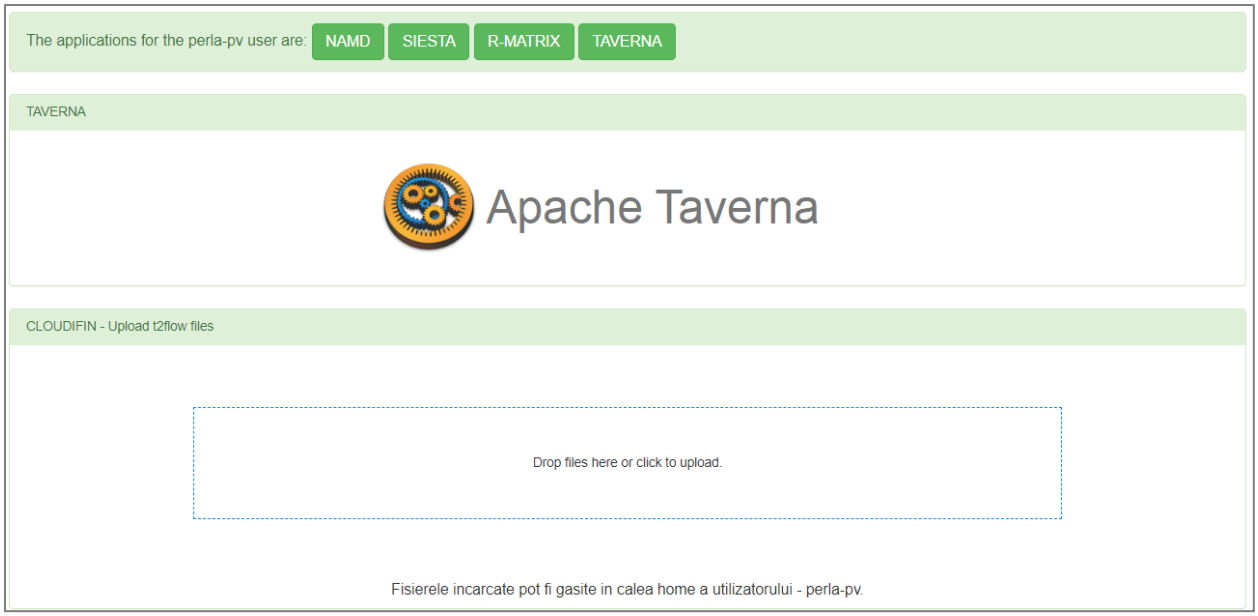


***Figura 4.3**: Pagina de incarcare a fisierelor t2flow*

Sistemul permite incarcarea fisierelor de input tipice Big Data, de mari dimensiuni, fiind testat cu fisiere de ordinul 5GB. Pentru aceasta, se apasa pe butonul "Upload Large File" din bara de comenzi reprezentata in Fig. 4.2, pentru a deschide pagina de mai jos.

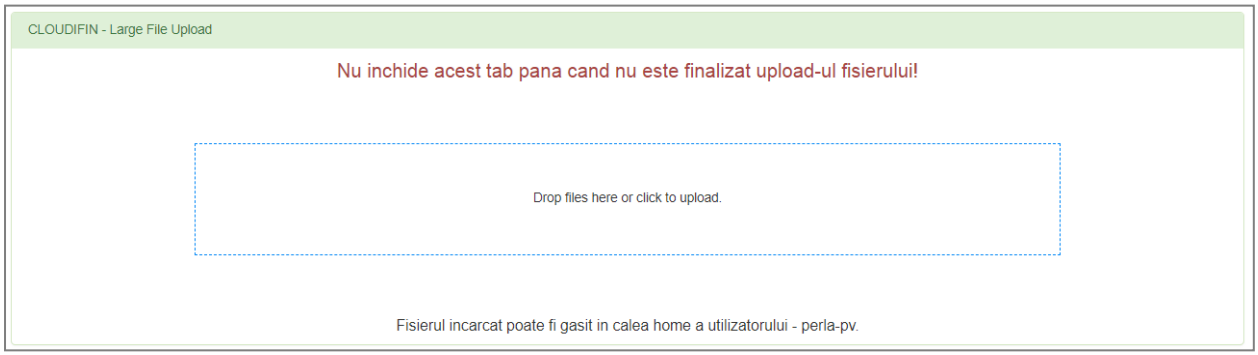


***Figura 4.4**: Pagina de incarcare a fisierelor Big Data*

4.2 Executia fluxurilor de lucru

Interfata grafica prin intermediul careia utilizatorul isi administreaza fluxurile de lucru este reprodusa in Fig. 4.5.

Interfata ii permite utilizatorului sa initieze asupra fisierelor listate urmatoarele operatiuni:

- vizualizarea continutului fisierului (buton *View*),
- executarea fisierelor executabile (*Execute*),
- stergerea fisierului (*Delete*),
- descarcare a fisierului pe calculatorul personal (*Downoad*).

Pentru rularea unui fisier t2flow se apasa butonul "Execute", iar fisierul este analizat in vederea identificarii datelor de intrare (*input*-uri) necesare rularii workflow-ului. Datele de intrare necesare sunt detectate de platforma iar utilizatorului nu-i mai ramane decat sa le completeze.

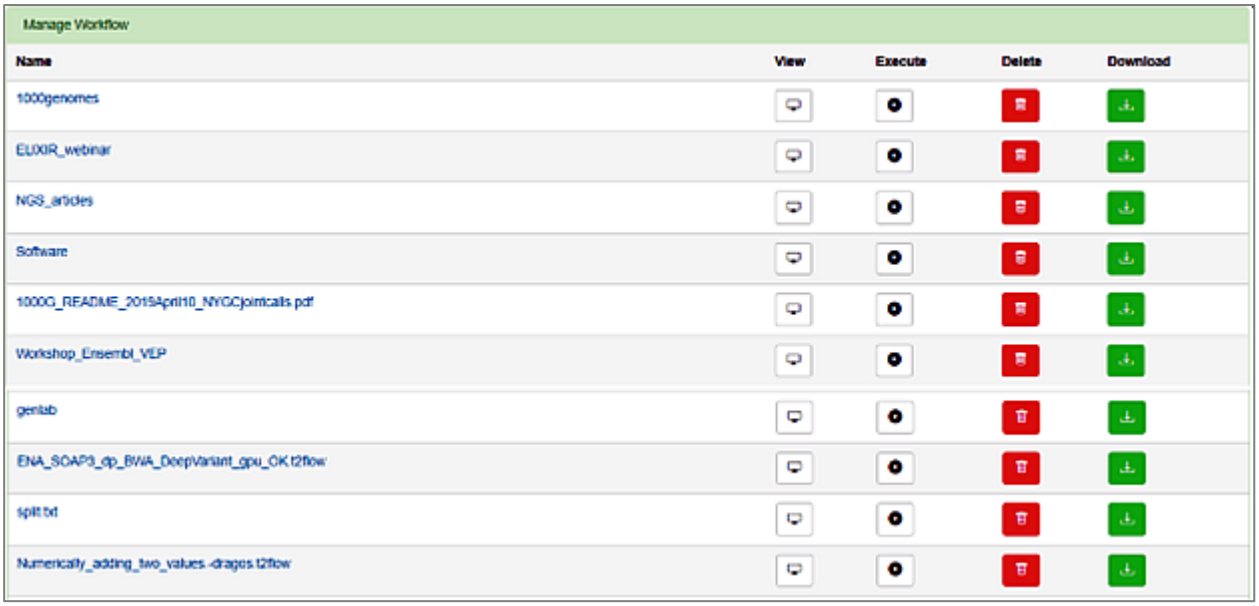


***Figura 4.5**: Pagina de administrare a fluxurilor de lucru*

In continuare sunt prezentate in detaliu procedurile programate in acest scop.

Din fisierele t2flow platforma extrage sirurile de caractere care sunt incadrate intre marcajele `<port>` si `</port>`.

```
<port><name>input_dir</name><depth>0</depth><granularDepth>0</granularDepth><ann
otations/></port>
<port><name>output_dir</name><depth>0</depth><granularDepth>0</granularDepth><an
notations/></port>
<port><name>fastq1</name><depth>0</depth><granularDepth>0</granularDepth><annota
tions/></port>
<port><name>ref_hgen</name><depth>0</depth><granularDepth>0</granularDepth><anno
tations/></port>
<port><name>maptype</name><depth>0</depth><granularDepth>0</granularDepth><annot
ations/></port>
<port><name>cpu</name><depth>0</depth><granularDepth>0</granularDepth><annotatio
ns/></port>
```

In acesta portiune a codului sunt definite datele de intrare (*input*-uri). Dupa aplicarea unor filtre sunt identificate urmatoarele date de intrare: `'input_dir'`, `'output_dir'`, `'fastq1'`, `'ref_hgen'`, `'maptype'`, `'cpu'`.

Aceste input-uri sunt afisate pe pagina web, iar dupa completarea lor utilizatorul apasa butonul "*Execute*" de lansare a workflow-ului din interfata (Fig.4.6). Dupa apasarea butonului, este generat mai intai un mesaj MQTT care este apoi este transmis, de exemplu, pe topicul "`cloud/req/cloudifin/gpu01`" catre agentul care asculta la acest topic. Mesajele care contin in topic stringul "`req`" sunt destinate agentilor, iar ceve care contin stringul "`res`" sunt receptionate de platforma. Din acest topic "`cloud/req/cloudifin/gpu01`" rezulta ca mesajul MQTT este transmis pe un agent (pentru ca regasim "`req`"), este destinat site-ului cloud "CLOUDIFIN" iar procesarea workflow-ului se realizeaza pe masina virtuala care se numeste `gpu01`.

Dupa completarea input-urilor (denumite *inputports* in *Taverna Workbench*), acestea devin:

```
-inputvalue input_dir "SRR099988"
-inputvalue output_dir "SRR099988"
-inputvalue fastq1 "SRR099988"
```

```
-inputvalue ref_hgen "Homo_sapiens_assembly19"
-inputvalue maptype "gase" -inputvalue cpu "1".
```

Mesajul MQTT generat de platforma dupa completarea input-urilor, numelui de utilizator (`username`) si fisierului selectat de acesta (`t2flow_name`) arata astfel:

```
-topic: "cloud/req/cloudifin/gpu01"
-mesaj:  username + "|/opt/taverna-workbench/executeworkflow
/nfs3/{{username}}/CECBID/" + t2flow_name + comp
```

unde `comp` sunt input-urile completate de utilizator de mai sus.

Mesajul MQTT este receptionat de agentul de pe masina virtuala cu numele *gpu01* si este impartit in doua parti prin folosiere delimitatorului "`|`"

```
username + "|/opt/taverna-workbench/executeworkflow /nfs3/{{username}}/CECBID/"
+ t2flow_name + comp
```

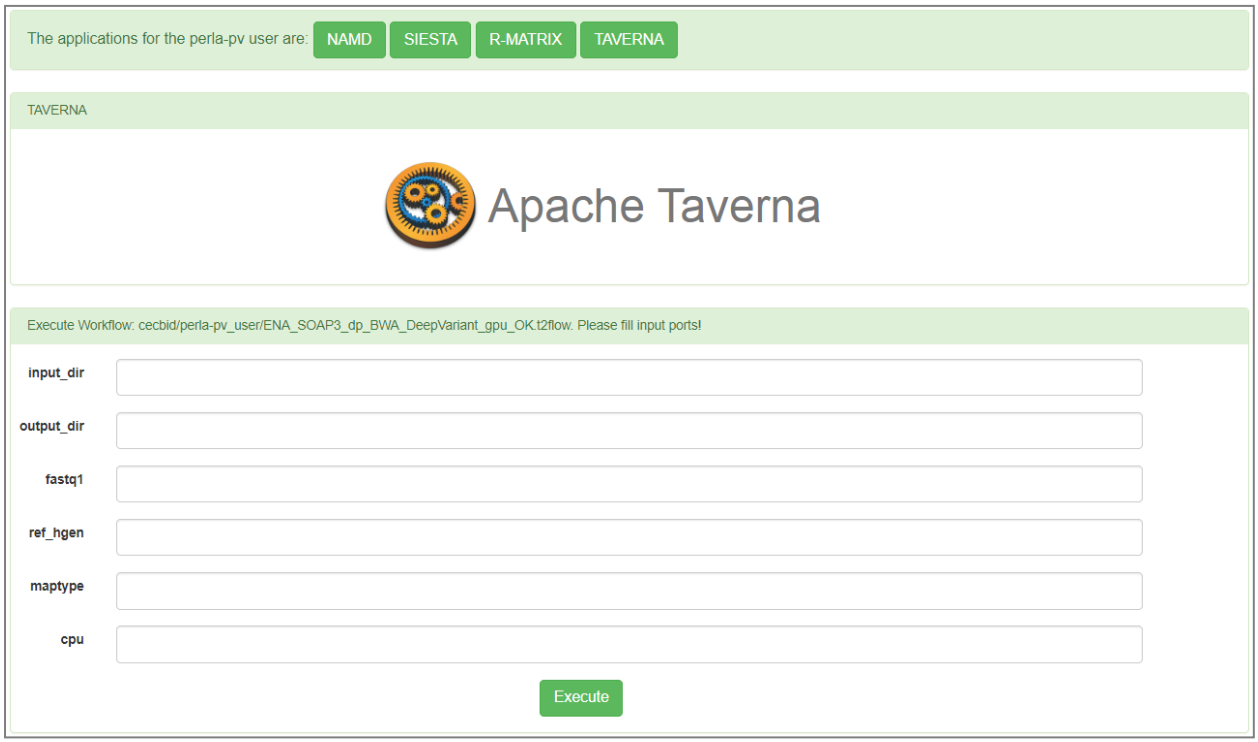


***Figura 4.6**: Pagina de specificare a input-urilor*

Daca se doreste acesarea consolei masinii virtuale se apasa butonul de "Console Access" din Fig. 4.2. Se deschide fereastra din Fig. 4.7 pentru selectarea resursei dorite.

Dupa apasarea butonului "Request Console" in lista de console va aparea link catre aceasta resursa.

***Figura 4.7**: Pagina de accesare a consolei unei masini virtuale*

5. Concluzii

- Toate fluxurile de lucru pentru analiza seturilor de date provenite din ambele linii genetice (germinativa si somatica) au rulat cu succes.
- In urma ultimelor modificări aduse etapei de pre-analiza a fluxurilor de lucru, timpul de rulare al analizei NGS a fost scurtat suplimentar cu 1-2 ore. Aceste modificări au permis automatizarea completa a procedurilor bioinformatice si integrarea fluxurilor de lucru in platforma CLOUDIFIN de analiza a datelor NGS.
- La finalul proiectului au fost obținute 2 fluxuri de lucru optimizate si validate riguros pentru analiza liniei germinative a datelor NGS umane si 3 fluxuri de lucru pentru analiza somatica, realizata in tandem din probe de secventiere normale si tumorale. Fluxurile de lucru dezvoltate in cadrul proiectului permit automatizarea principalelor tipuri de proceduri bioinformatice pentru analiza seturilor de date NGS (WGS/WES) umane.
- Cele 5 fluxuri de lucru au fost integrate in platforma software dezvoltata pentru analiza fisierelor de date de mari dimensiuni (Big Data) provenite din secventierea de noua generatie (NGS).
- S-a dezvoltat si implementat o interfata grafica de acces al utilizatorilor autorizati la serviciile platformei de analiza a datelor NGS.

Referinte

[1] Wolstencroft K, Haines R, Fellows D, Williams A, Withers D, Owen S, et al. The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud. Nucleic Acids Res 2013;41:W557–61. https://doi.org/10.1093/nar/gkt328.

[2] McKenna A, Hanna M, Banks E, Sivachenko A, Cibulskis K, Kernytsky A, et al. The Genome Analysis Toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. Genome Res 2010;20:1297–303. https://doi.org/10.1101/gr.107524.110.

[3] Li H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. ArXiv:13033997 [q-Bio] 2013.

[4] Poplin R, Chang P-C, Alexander D, Schwartz S, Colthurst T, Ku A, et al. A universal SNP and small-indel variant caller using deep neural networks. Nat Biotechnol 2018;36:983–7. https://doi.org/10.1038/nbt.4235.

[5] Benjamin D, Sato T, Cibulskis K, Getz G, Stewart C, Lichtenstein L. Calling Somatic SNVs and Indels with Mutect 2019. https://doi.org/10.1101/861054.

[6] Kim S, Scheffler K, Halpern AL, Bekritsky MA, Noh E, Källberg M, et al. Strelka2: fast and accurate calling of germline and somatic variants. Nat Methods 2018;15:591–4. https://doi.org/10.1038/s41592-018-0051-x.

[7] Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, et al. The Sequence Alignment/Map format and SAMtools. Bioinformatics 2009;25:2078–9. https://doi.org/10.1093/bioinformatics/btp352.

[8] Danecek P, Bonfield JK, Liddle J, Marshall J, Ohan V, Pollard MO, et al. Twelve years of SAMtools and BCFtools. Gigascience 2021;10:giab008. https://doi.org/10.1093/gigascience/giab008.

[9] Picard Tools - By Broad Institute n.d. http://broadinstitute.github.io/picard/ (accessed November 28, 2020).

[10] Poplin R, Ruano-Rubio V, DePristo MA, Fennell TJ, Carneiro MO, Auwera GAV der, et al. Scaling accurate genetic variant discovery to tens of thousands of samples. BioRxiv 2018:201178. https://doi.org/10.1101/201178.

[11] Cingolani P, Platts A, Wang LL, Coon M, Nguyen T, Wang L, et al. A program for annotating and predicting the effects of single nucleotide polymorphisms, SnpEff: SNPs in the genome of Drosophila melanogaster strain w1118; iso-2; iso-3. Fly (Austin) 2012;6:80–92. https://doi.org/10.4161/fly.19695.

[12] Cingolani P, Patel VM, Coon M, Nguyen T, Land SJ, Ruden DM, et al. Using Drosophila melanogaster as a Model for Genotoxic Chemical Mutational Studies with a New Program, SnpSift. Front Genet 2012;3:35. https://doi.org/10.3389/fgene.2012.00035.

[13] Landrum MJ, Lee JM, Benson M, Brown GR, Chao C, Chitipiralla S, et al. ClinVar: improving access to variant interpretations and supporting evidence. Nucleic Acids Res 2018;46:D1062–7. https://doi.org/10.1093/nar/gkx1153.

[14] Szegedy C, Ioffe S, Vanhoucke V, Alemi A. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. ArXiv:160207261 [Cs] 2016.

[15] Savage P, Pacis A, Kuasne H, Liu L, Lai D, Wan A, et al. Chemogenomic profiling of breast cancer patient-derived xenografts reveals targetable vulnerabilities for difficult-to-treat tumors. Commun Biol 2020;3:310. https://doi.org/10.1038/s42003-020-1042-x.

[16] Genome in a Bottle. NIST 2012.

[17] Eberle MA, Fritzilas E, Krusche P, Källberg M, Moore BL, Bekritsky MA, et al. A reference data set of 5.4 million phased human variants validated by genetic inheritance from sequencing a three-generation 17-member pedigree. Genome Res 2017;27:157–64. https://doi.org/10.1101/gr.210500.116.

[18] Supernat A, Vidarsson OV, Steen VM, Stokowy T. Comparison of three variant callers for human whole genome sequencing. Sci Rep 2018;8:17851. https://doi.org/10.1038/s41598-018-36177-7.

[19] Li H. lh3/CHM-eval 2023.

[20] Haplotype Comparison Tools 2023.

[21] Barbitoff YA, Abasov R, Tvorogova VE, Glotov AS, Predeus AV. Systematic benchmark of state-of-the-art variant calling pipelines identifies major factors affecting accuracy of coding sequence variant discovery. BMC Genomics 2022;23:155. https://doi.org/10.1186/s12864-022-08365-3.

[22] Shand M, Soto J, Lichtenstein L, Benjamin D, Farjoun Y, Brody Y, et al. A validated lineage-derived somatic truth data set enables benchmarking in cancer genome analysis. Commun Biol 2020;3:744. https://doi.org/10.1038/s42003-020-01460-9.

[23] Lee AY, Ewing AD, Ellrott K, Hu Y, Houlahan KE, Bare JC, et al. Combining accurate tumor genome simulation with crowdsourcing to benchmark somatic structural variant detection. Genome Biol 2018;19:188. https://doi.org/10.1186/s13059-018-1539-5.

[24] Mu JC, Mohiyuddin M, Li J, Bani Asadi N, Gerstein MB, Abyzov A, et al. VarSim: a high-fidelity simulation and validation framework for high-throughput genome sequencing with cancer applications. Bioinformatics 2015;31:1469–71. https://doi.org/10.1093/bioinformatics/btu828.

[25] Tate JG, Bamford S, Jubb HC, Sondka Z, Beare DM, Bindal N, et al. COSMIC: the Catalogue Of Somatic Mutations In Cancer. Nucleic Acids Res 2019;47:D941–7. https://doi.org/10.1093/nar/gky1015.

[26] CNV-Sim by NabaviLab n.d. https://nabavilab.github.io/CNV-Sim/ (accessed November 27, 2021).

[27] N.d. https://s3.amazonaws.com/biodata (accessed November 30, 2021).

[28] Bolger AM, Lohse M, Usadel B. Trimmomatic: a flexible trimmer for Illumina sequence data. Bioinformatics 2014;30:2114–20. https://doi.org/10.1093/bioinformatics/btu170.

[29] Chen X, Schulz-Trieglaff O, Shaw R, Barnes B, Schlesinger F, Källberg M, et al. Manta: rapid detection of structural variants and indels for germline and cancer sequencing applications. Bioinformatics 2016;32:1220–2. https://doi.org/10.1093/bioinformatics/btv710.

[30] Cleary JG, Braithwaite R, Gaastra K, Hilbush B, Inglis S, Irvine SA, et al. Comparing Variant Call Files for Performance Benchmarking of Next-Generation Sequencing Variant Calling Pipelines 2015. https://doi.org/10.1101/023754.

[31] Luo R, Wong T, Zhu J, Liu C-M, Zhu X, Wu E, et al. SOAP3-dp: fast, accurate and sensitive GPU-based short read aligner. PLoS One 2013;8:e65632. https://doi.org/10.1371/journal.pone.0065632.

[32] Ahmed N, Lévy J, Ren S, Mushtaq H, Bertels K, Al-Ars Z. GASAL2: a GPU accelerated sequence alignment library for high-throughput NGS data. BMC Bioinformatics 2019;20:520. https://doi.org/10.1186/s12859-019-3086-9.

[33] Burrows M, Wheeler DJ. A block-sorting lossless data compression algorithm. 1994.

6. ANEXA

Codul sursa pentru fluxul de lucru ENA_SOAP3_dp_BWA_DeepVariant_gpu_OK.t2flow

<workflow xmlns="http://taverna.sf.net/2008/xml/t2flow" version="1" producedBy="taverna-core-2.5.0"><dataflow id="28255bca-da42-45ce-87ed-81fa90c33568" role="top"><name>Workflow1</name><inputPorts><port><name>input_dir</name><depth>0</depth><granularDepth>0</granularDepth><annotations />></port><port><name>output_dir</name><depth>0</depth><granularDepth>0</granularDepth><annotations />></port><port><name>fastq1</name><depth>0</depth><granularDepth>0</granularDepth><annotations />></port><port><name>ref_hgen</name><depth>0</depth><granularDepth>0</granularDepth><annotations />></port><port><name>maptype</name><depth>0</depth><granularDepth>0</granularDepth><annotations />></port><port><name>cpu</name><depth>0</depth><granularDepth>0</granularDepth><annotations />></port></inputPorts><outputPorts><port><name>ena_downloader</name><lastPredictedDepth>0</lastPredictedDepth><annotations />></port><port><name>mapping</name><lastPredictedDepth>0</lastPredictedDepth><annotations />></port><port><name>Deep_Variant</name><lastPredictedDepth>0</lastPredictedDepth><annotations />></port></outputPorts><processors><processor><name>mapping</name><inputPorts><port><name>ref_hgen</name><depth>0</depth></port><port><name>output_dir</name><depth>0</depth></port><port><name>input_dir</name><depth>0</depth></port><port><name>maptype</name><depth>0</depth></port><port><name>fastq1</name><depth>0</depth></port><port><name>cpu</name><depth>0</depth></port></inputPorts><outputPorts><port><name>STDOUT</name><depth>0</depth><granularDepth>0</granularDepth></port></outputPorts><annotations /><activities><activity><raven><group>net.sf.taverna.t2.activities</group><artifact>external-tool-activity</artifact><version>1.5</version></raven><class>net.sf.taverna.t2.activities.externaltool.ExternalToolActivity</class><inputMap><map from="ref_hgen" to="ref_hgen" /><map from="input_dir" to="input_dir" /><map from="output_dir" to="output_dir" /><map from="fastq1" to="fastq1" /><map from="cpu" to="cpu" /><map from="maptype" to="maptype" /></inputMap><outputMap><map from="STDOUT" to="STDOUT" /></outputMap><configBean encoding="xstream"><net.sf.taverna.t2.activities.externaltool.ExternalToolActivityConfigurationBean xmlns="">

<mechanismType>789663B8-DA91-428A-9F7D-B3F3DA185FD4</mechanismType>

<mechanismName>default local</mechanismName>

<mechanismXML><?xml version="1.0" encoding="UTF-8"?>

<localInvocation />

</mechanismXML>

<externaltoolid>e7b37999-1b40-4211-aed8-76d3ddc055f3</externaltoolid>

<useCaseDescription>

<usecaseid />

<description>fastq_1 = e.g. SRR099988

input_dir = e.g. SRR099988

output_dir = e.g. SRR099988

maptype = bwa, soap3 or gase

ref_gen = Homo_sapiens_assembly19 or Homo_sapiens_assembly38

</description>

<command>export input_dir=/nfs3/gnecula/CECBID/1000genomes/samples/%%input_dir%%

mkdir /nfs3/gnecula/CECBID/1000genomes/samples/%%output_dir%%/%%maptype%%_bam

export output_dir=/nfs3/gnecula/CECBID/1000genomes/samples/%%output_dir%%/%%maptype%%_bam

cd $output_dir

if ["%%maptype%%" == "soap3"]; then

export soap3dp_dir=/nfs3/gnecula/CECBID/Software/SOAP3-dp_cuda11_vm

export ref_dir=/nfs3/gnecula/CECBID/Software/SOAP3-dp_index

export picard=/nfs3/gnecula/CECBID/Software/picard.jar

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/cuda-11.0/targets/x86_64-linux/lib

$soap3dp_dir/soap3-dp pair $ref_dir/%%ref_hgen%%.fasta.index $input_dir/%%fastq1%%_1.fastq.gz $input_dir/%%fastq1%%_2.fastq.gz -c 0 -b 3 -o $output_dir/%%fastq1%% > $output_dir/%%fastq1%%.log

wait

samtools merge -@ %%cpu%% %%fastq1%%_merged.bam \

%%fastq1%%.gout.1 \

%%fastq1%%.gout.10 \

%%fastq1%%.gout.11 \

%%fastq1%%.gout.12 \

%%fastq1%%.gout.13 \

%%fastq1%%.gout.14 \

%%fastq1%%.gout.15 \

%%fastq1%%.gout.16 \

%%fastq1%%.gout.17 \

%%fastq1%%.gout.18 \

%%fastq1%%.gout.19 \

%%fastq1%%.gout.2 \

```
%%fastq1%%.gout.20 \
%%fastq1%%.gout.3 \
%%fastq1%%.gout.4 \
%%fastq1%%.gout.5 \
%%fastq1%%.gout.6 \
%%fastq1%%.gout.7 \
%%fastq1%%.gout.8 \
%%fastq1%%.gout.9 \
%%fastq1%%.dpout.1 \
%%fastq1%%.unpair

wait

samtools sort -@ %%cpu%% %%fastq1%%_merged.bam -o %%fastq1%%_sorted.bam
samtools index -@ %%cpu%% %%fastq1%%_sorted.bam %%fastq1%%_sorted.bam.bai

wait

nohup java $jvm_args -XX:ParallelGCThreads=%%cpu%% -jar $picard \
AddOrReplaceReadGroups \
I=%%fastq1%%_sorted.bam  \
O=%%fastq1%%_sorted_rg.bam \
SORT_ORDER=coordinate \
CREATE_INDEX=true \
RGPL=illumina \
RGLB=Lib1 \
RGPU=%%fastq1%% \
RGID=%%fastq1%% \
RGSM=%%fastq1%%

wait

else
    echo "soap3-dp mapping not selected"

fi
```

```
if [ "%%maptype%%" == "bwa" ]; then


export
ref_dir=/nfs3/gnecula/CECBID/GATK_somatic_workflow/GATK_somatic_germline/gatk_files/so
matic/ref

bwa mem -Y -K 100000000 -t %%cpu%% -R '@RG\tID:%%fastq1%%\tSM:%%fastq1%%'
$ref_dir/%%ref_hgen%%.fasta $input_dir/%%fastq1%%_1.fastq.gz
$input_dir/%%fastq1%%_2.fastq.gz | samtools view -Shb -o $output_dir/%%fastq1%%.bam

samtools sort -@ %%cpu%% %%fastq1%%.bam &gt; %%fastq1%%_sorted_rg.bam

samtools index -@ %%cpu%% %%fastq1%%_sorted_rg.bam
%%fastq1%%_sorted_rg.bam.bai


else

   echo "bwa mapping not selected"


fi


if [ "%%maptype%%" == "gase" ]; then


export gase_bin=/nfs3/gnecula/CECBID/Software/gase-gasal2/gase

export
ref_dir=/nfs3/gnecula/CECBID/GATK_somatic_workflow/GATK_somatic_germline/gatk_files/so
matic/ref


$gase_bin gase_aln -l 150 -t %%cpu%% -Y -R '@RG\tID:%%fastq1%%\tSM:%%fastq1%%'
$ref_dir/%%ref_hgen%%.fasta $input_dir/%%fastq1%%_1.fastq.gz
$input_dir/%%fastq1%%_2.fastq.gz &gt; $output_dir/%%fastq1%%.sam

samtools view -Shb -@ %%cpu%% %%fastq1%%.sam &gt; %%fastq1%%.bam

samtools sort -@ %%cpu%% %%fastq1%%.bam &gt; %%fastq1%%_sorted_rg.bam

samtools index -@ %%cpu%% %%fastq1%%_sorted_rg.bam
%%fastq1%%_sorted_rg.bam.bai


else

   echo "gase mapping not selected"


fi

</command>

   <preparingTimeoutInSeconds>1200</preparingTimeoutInSeconds>

   <executionTimeoutInSeconds>1800</executionTimeoutInSeconds>

   <tags>

     <string>cpu</string>
```

```
    <string>fastq1</string>
    <string>input_dir</string>
    <string>maptype</string>
    <string>output_dir</string>
    <string>ref_hgen</string>
  </tags>
  <REs />
  <queue__preferred />
  <queue__deny />
  <static__inputs />
  <inputs>
    <entry>
      <string>ref_hgen</string>
      <de.uni__luebeck.inb.knowarc.usecases.ScriptInputUser>
        <tag>ref_hgen</tag>
        <file>false</file>
        <tempFile>false</tempFile>
        <binary>false</binary>
        <charsetName>windows-1252</charsetName>
        <forceCopy>false</forceCopy>
        <list>false</list>
        <concatenate>false</concatenate>
        <mime />
      </de.uni__luebeck.inb.knowarc.usecases.ScriptInputUser>
    </entry>
    <entry>
      <string>output_dir</string>
      <de.uni__luebeck.inb.knowarc.usecases.ScriptInputUser>
        <tag>output_dir</tag>
        <file>false</file>
        <tempFile>false</tempFile>
        <binary>false</binary>
        <charsetName>windows-1252</charsetName>
        <forceCopy>false</forceCopy>
        <list>false</list>
        <concatenate>false</concatenate>
        <mime />
      </de.uni__luebeck.inb.knowarc.usecases.ScriptInputUser>
```

```
</entry>
<entry>
  <string>input_dir</string>
  <de.uni__luebeck.inb.knowarc.usecases.ScriptInputUser>
    <tag>input_dir</tag>
    <file>false</file>
    <tempFile>false</tempFile>
    <binary>false</binary>
    <charsetName>windows-1252</charsetName>
    <forceCopy>false</forceCopy>
    <list>false</list>
    <concatenate>false</concatenate>
    <mime />
  </de.uni__luebeck.inb.knowarc.usecases.ScriptInputUser>
</entry>
<entry>
  <string>fastq1</string>
  <de.uni__luebeck.inb.knowarc.usecases.ScriptInputUser>
    <tag>fastq1</tag>
    <file>false</file>
    <tempFile>false</tempFile>
    <binary>false</binary>
    <charsetName>windows-1252</charsetName>
    <forceCopy>false</forceCopy>
    <list>false</list>
    <concatenate>false</concatenate>
    <mime />
  </de.uni__luebeck.inb.knowarc.usecases.ScriptInputUser>
</entry>
<entry>
  <string>cpu</string>
  <de.uni__luebeck.inb.knowarc.usecases.ScriptInputUser>
    <tag>cpu</tag>
    <file>false</file>
    <tempFile>false</tempFile>
    <binary>false</binary>
    <charsetName>windows-1252</charsetName>
    <forceCopy>false</forceCopy>
```

```
        <list>false</list>
        <concatenate>false</concatenate>
        <mime />
       </de.uni__luebeck.inb.knowarc.usecases.ScriptInputUser>
      </entry>
      <entry>
       <string>maptype</string>
       <de.uni__luebeck.inb.knowarc.usecases.ScriptInputUser>
        <tag>maptype</tag>
        <file>false</file>
        <tempFile>false</tempFile>
        <binary>false</binary>
        <charsetName>windows-1252</charsetName>
        <forceCopy>false</forceCopy>
        <list>false</list>
        <concatenate>false</concatenate>
        <mime />
       </de.uni__luebeck.inb.knowarc.usecases.ScriptInputUser>
      </entry>
    </inputs>
    <outputs />
    <includeStdIn>false</includeStdIn>
    <includeStdOut>true</includeStdOut>
    <includeStdErr>true</includeStdErr>
    <validReturnCodes>
     <int>0</int>
    </validReturnCodes>
   </useCaseDescription>
   <edited>false</edited>
</net.sf.taverna.t2.activities.externaltool.ExternalToolActivityConfigurationBean></configBean><annotations
/></activity></activities><dispatchStack><dispatchLayer><raven><group>net.sf.taverna.t2.core</group><artifact>workflowmodel-
impl</artifact><version>1.5</version></raven><class>net.sf.taverna.t2.workflowmodel.processor.dispatch.layers.Parallelize</class><configBean
encoding="xstream"><net.sf.taverna.t2.workflowmodel.processor.dispatch.layers.ParallelizeConfig xmlns="">
   <maxJobs>1</maxJobs>
</net.sf.taverna.t2.workflowmodel.processor.dispatch.layers.ParallelizeConfig></configBean>
</dispatchLayer><dispatchLayer><raven><group>net.sf.taverna.t2.core</group><artifact>
workflowmodel-
```

impl</artifact><version>1.5</version></raven><class>net.sf.taverna.t2.workflowmodel.processor.dispatch.layers.ErrorBounce</class><configBean encoding="xstream"><null xmlns="" /></configBean></dispatchLayer><dispatchLayer><raven><group>net.sf.taverna.t2.core</group><artifact>workflowmodel-
impl</artifact><version>1.5</version></raven><class>net.sf.taverna.t2.workflowmodel.processor.dispatch.layers.Failover</class><configBean encoding="xstream"><null xmlns="" /></configBean></dispatchLayer><dispatchLayer><raven><group>net.sf.taverna.t2.core</group><artifact>workflowmodel-
impl</artifact><version>1.5</version></raven><class>net.sf.taverna.t2.workflowmodel.processor.dispatch.layers.Retry</class><configBean
encoding="xstream"><net.sf.taverna.t2.workflowmodel.processor.dispatch.layers.RetryConfig xmlns="">

<backoffFactor>1.0</backoffFactor>

<initialDelay>1000</initialDelay>

<maxDelay>5000</maxDelay>

<maxRetries>0</maxRetries>

</net.sf.taverna.t2.workflowmodel.processor.dispatch.layers.RetryConfig></configBean></dispatchLayer><dispatchLayer><raven><group>net.sf.taverna.t2.core</group><artifact>workflowmodel-
impl</artifact><version>1.5</version></raven><class>net.sf.taverna.t2.workflowmodel.processor.dispatch.layers.Invoke</class><configBean encoding="xstream"><null xmlns="" /></configBean></dispatchLayer></dispatchStack><iterationStrategyStack><iteration><strategy><cross><port name="ref_hgen" depth="0" /><port name="output_dir" depth="0" /><port name="input_dir" depth="0" /><port name="maptype" depth="0" /><port name="fastq1" depth="0" /><port name="cpu" depth="0" /></cross></strategy></iteration></iterationStrategyStack></processor><processor><name>Deep_Variant</name><inputPorts><port><name>output_dir</name><depth>0</depth></port><port><name>ref_hgen</name><depth>0</depth></port><port><name>maptype</name><depth>0</depth></port><port><name>fastq1</name><depth>0</depth></port><port><name>cpu</name><depth>0</depth></port></inputPorts><outputPorts><port><name>STDOUT</name><depth>0</depth><granularDepth>0</granularDepth></port></outputPorts><annotations
/><activities><activity><raven><group>net.sf.taverna.t2.activities</group><artifact>external-tool-
activity</artifact><version>1.5</version></raven><class>net.sf.taverna.t2.activities.externaltool.ExternalToolActivity</class><inputMap><map from="ref_hgen" to="ref_hgen" /><map from="output_dir" to="output_dir" /><map from="fastq1" to="fastq1" /><map from="cpu" to="cpu" /><map from="maptype" to="maptype" /></inputMap><outputMap><map from="STDOUT" to="STDOUT" /></outputMap><configBean encoding="xstream"><net.sf.taverna.t2.activities.externaltool.ExternalToolActivityConfigurationBean xmlns="">

<mechanismType>789663B8-DA91-428A-9F7D-B3F3DA185FD4</mechanismType>

<mechanismName>default local</mechanismName>

<mechanismXML><?xml version="1.0" encoding="UTF-8"?>

<localInvocation />

</mechanismXML>

<externaltoolid>d1941d40-5b97-458c-93a6-c8934adead8c</externaltoolid>

<useCaseDescription>

<usecaseid />

```
    <description />

<command>INPUT_DIR="/nfs3/gnecula/CECBID/1000genomes/samples/%%output_dir%%/
%%maptype%%_bam"
INPUT_DIR_REF="/nfs3/gnecula/CECBID/GATK_somatic_workflow/GATK_somatic_germline/ga
tk_files/somatic/ref"
OUTPUT_DIR="/nfs3/gnecula/CECBID/1000genomes/samples/%%output_dir%%/out_DV"
BIN_VERSION="1.3.0"

###  --regions "chr20:10,000,000-10,010,000" \

sudo docker run --gpus 1 \
  -v "${INPUT_DIR}":"/input" \
  -v "${INPUT_DIR_REF}":"/input_ref" \
  -v "${OUTPUT_DIR}":"/output" \
  google/deepvariant:"${BIN_VERSION}-gpu" \
  /opt/deepvariant/bin/run_deepvariant \
  --model_type=WGS \
  --ref=/input_ref/%%ref_hgen%%.fasta \
  --reads=/input/%%fastq1%%_sorted_rg.bam \
  --output_vcf=/output/output.vcf.gz \
  --output_gvcf=/output/output.g.vcf.gz \
  --intermediate_results_dir=/output/tmp \
  --num_shards=%%cpu%%

wait</command>
    <preparingTimeoutInSeconds>1200</preparingTimeoutInSeconds>
    <executionTimeoutInSeconds>1800</executionTimeoutInSeconds>
    <tags>
      <string>cpu</string>
      <string>fastq1</string>
      <string>maptype</string>
      <string>output_dir</string>
      <string>ref_hgen</string>
    </tags>
    <REs />
    <queue__preferred />
    <queue__deny />
    <static__inputs />
```

```
<inputs>
  <entry>
    <string>ref_hgen</string>
    <de.uni__luebeck.inb.knowarc.usecases.ScriptInputUser>
      <tag>ref_hgen</tag>
      <file>false</file>
      <tempFile>false</tempFile>
      <binary>false</binary>
      <charsetName>windows-1252</charsetName>
      <forceCopy>false</forceCopy>
      <list>false</list>
      <concatenate>false</concatenate>
      <mime />
    </de.uni__luebeck.inb.knowarc.usecases.ScriptInputUser>
  </entry>
  <entry>
    <string>output_dir</string>
    <de.uni__luebeck.inb.knowarc.usecases.ScriptInputUser>
      <tag>output_dir</tag>
      <file>false</file>
      <tempFile>false</tempFile>
      <binary>false</binary>
      <charsetName>windows-1252</charsetName>
      <forceCopy>false</forceCopy>
      <list>false</list>
      <concatenate>false</concatenate>
      <mime />
    </de.uni__luebeck.inb.knowarc.usecases.ScriptInputUser>
  </entry>
  <entry>
    <string>fastq1</string>
    <de.uni__luebeck.inb.knowarc.usecases.ScriptInputUser>
      <tag>fastq1</tag>
      <file>false</file>
      <tempFile>false</tempFile>
      <binary>false</binary>
      <charsetName>windows-1252</charsetName>
      <forceCopy>false</forceCopy>
```

```
      <list>false</list>
      <concatenate>false</concatenate>
      <mime />
    </de.uni__luebeck.inb.knowarc.usecases.ScriptInputUser>
  </entry>
  <entry>
    <string>cpu</string>
    <de.uni__luebeck.inb.knowarc.usecases.ScriptInputUser>
      <tag>cpu</tag>
      <file>false</file>
      <tempFile>false</tempFile>
      <binary>false</binary>
      <charsetName>windows-1252</charsetName>
      <forceCopy>false</forceCopy>
      <list>false</list>
      <concatenate>false</concatenate>
      <mime />
    </de.uni__luebeck.inb.knowarc.usecases.ScriptInputUser>
  </entry>
  <entry>
    <string>maptype</string>
    <de.uni__luebeck.inb.knowarc.usecases.ScriptInputUser>
      <tag>maptype</tag>
      <file>false</file>
      <tempFile>false</tempFile>
      <binary>false</binary>
      <charsetName>windows-1252</charsetName>
      <forceCopy>false</forceCopy>
      <list>false</list>
      <concatenate>false</concatenate>
      <mime />
    </de.uni__luebeck.inb.knowarc.usecases.ScriptInputUser>
  </entry>
</inputs>
<outputs />
<includeStdIn>false</includeStdIn>
<includeStdOut>true</includeStdOut>
<includeStdErr>true</includeStdErr>
```

```
    <validReturnCodes>
      <int>0</int>
    </validReturnCodes>
  </useCaseDescription>
  <edited>false</edited>
</net.sf.taverna.t2.activities.externaltool.ExternalToolActivityConfigurationBean></configBean><annotations
/></activity></activities><dispatchStack><dispatchLayer><raven><group>net.sf.taverna.t2.core</group><artifact>workflowmodel-
impl</artifact><version>1.5</version></raven><class>net.sf.taverna.t2.workflowmodel.processor.dispatch.layers.Parallelize</class><configBean
encoding="xstream"><net.sf.taverna.t2.workflowmodel.processor.dispatch.layers.ParallelizeConfig xmlns="">
  <maxJobs>1</maxJobs>
</net.sf.taverna.t2.workflowmodel.processor.dispatch.layers.ParallelizeConfig></configBean></dispatchLayer><dispatchLayer><raven><group>net.sf.taverna.t2.core</group><artifact>workflowmodel-
impl</artifact><version>1.5</version></raven><class>net.sf.taverna.t2.workflowmodel.processor.dispatch.layers.ErrorBounce</class><configBean encoding="xstream"><null xmlns="" /></configBean></dispatchLayer><dispatchLayer><raven><group>net.sf.taverna.t2.core</group><artifact>workflowmodel-
impl</artifact><version>1.5</version></raven><class>net.sf.taverna.t2.workflowmodel.processor.dispatch.layers.Failover</class><configBean encoding="xstream"><null xmlns="" /></configBean></dispatchLayer><dispatchLayer><raven><group>net.sf.taverna.t2.core</group><artifact>workflowmodel-
impl</artifact><version>1.5</version></raven><class>net.sf.taverna.t2.workflowmodel.processor.dispatch.layers.Retry</class><configBean
encoding="xstream"><net.sf.taverna.t2.workflowmodel.processor.dispatch.layers.RetryConfig xmlns="">
  <backoffFactor>1.0</backoffFactor>
  <initialDelay>1000</initialDelay>
  <maxDelay>5000</maxDelay>
  <maxRetries>0</maxRetries>
</net.sf.taverna.t2.workflowmodel.processor.dispatch.layers.RetryConfig></configBean></dispatchLayer><dispatchLayer><raven><group>net.sf.taverna.t2.core</group><artifact>workflowmodel-
impl</artifact><version>1.5</version></raven><class>net.sf.taverna.t2.workflowmodel.processor.dispatch.layers.Invoke</class><configBean encoding="xstream"><null xmlns="" /></configBean></dispatchLayer></dispatchStack><iterationStrategyStack><iteration><strategy><cross><port name="output_dir" depth="0" /><port name="ref_hgen" depth="0" /><port name="maptype" depth="0" /><port name="fastq1" depth="0" /><port name="cpu" depth="0"
/></cross></strategy></iteration></iterationStrategyStack></processor><processor><name>ena_downloader</name><inputPorts><port><name>input_dir</name><depth>0</depth></port><port><name>fastq1</name><depth>0</depth></port></inputPorts><outputPorts><port><name>STDOUT</name><depth>0</depth><granularDepth>0</granularDepth></port></outputPorts><annotations
/><activities><activity><raven><group>net.sf.taverna.t2.activities</group><artifact>external-tool-
activity</artifact><version>1.5</version></raven><class>net.sf.taverna.t2.activities.extern
```

altool.ExternalToolActivity</class><inputMap><map from="input_dir" to="input_dir" /><map from="fastq1" to="fastq1" /></inputMap><outputMap><map from="STDOUT" to="STDOUT" /></outputMap><configBean encoding="xstream"><net.sf.taverna.t2.activities.externaltool.ExternalToolActivityConfigurationBean xmlns="">

<mechanismType>789663B8-DA91-428A-9F7D-B3F3DA185FD4</mechanismType>

<mechanismName>default local</mechanismName>

<mechanismXML><?xml version="1.0" encoding="UTF-8"?>

<localInvocation />

</mechanismXML>

<externaltoolid>ae031485-b926-4d42-aaec-a6070cc2fda3</externaltoolid>

<useCaseDescription>

<usecaseid />

<description>mkdir /nfs3/gnecula/CECBID/1000genomes/samples/ERR949836

java -jar ena-file-downloader.jar --accessions=ERR949836 --format=READS_FASTQ --location=/nfs3/gnecula/CECBID/1000genomes/samples/ERR949836 --protocol=FTP --asperaLocation=null

mv /nfs3/gnecula/CECBID/1000genomes/samples/ERR949836/reads_fastq/ERR949836/*.fastq.gz /nfs3/gnecula/CECBID/1000genomes/samples/ERR949836

rm -rf /nfs3/gnecula/CECBID/1000genomes/samples/ERR949836/reads_fastq

</description>

<command>export ena_down=/nfs3/gnecula/CECBID/Software/ena_downloader/ena-file-downloader.jar

mkdir /nfs3/gnecula/CECBID/1000genomes/samples/%%input_dir%%

export input_dir=/nfs3/gnecula/CECBID/1000genomes/samples/%%input_dir%%

java -jar $ena_down --accessions=%%fastq1%% --format=READS_FASTQ --location=$input_dir --protocol=FTP --asperaLocation=null

mv /nfs3/gnecula/CECBID/1000genomes/samples/%%input_dir%%/reads_fastq/%%fastq1%%/*.fastq.gz /nfs3/gnecula/CECBID/1000genomes/samples/%%input_dir%%

rm -rf /nfs3/gnecula/CECBID/1000genomes/samples/%%input_dir%%/reads_fastq

#gunzip /nfs3/gnecula/CECBID/1000genomes/samples/%%input_dir%%/*.fastq.gz

wait

</command>

<preparingTimeoutInSeconds>1200</preparingTimeoutInSeconds>

<executionTimeoutInSeconds>1800</executionTimeoutInSeconds>

```
<tags>
  <string>fastq1</string>
  <string>input_dir</string>
</tags>
<REs />
<queue__preferred />
<queue__deny />
<static__inputs />
<inputs>
  <entry>
    <string>input_dir</string>
    <de.uni__luebeck.inb.knowarc.usecases.ScriptInputUser>
      <tag>input_dir</tag>
      <file>false</file>
      <tempFile>false</tempFile>
      <binary>false</binary>
      <charsetName>windows-1252</charsetName>
      <forceCopy>false</forceCopy>
      <list>false</list>
      <concatenate>false</concatenate>
      <mime />
    </de.uni__luebeck.inb.knowarc.usecases.ScriptInputUser>
  </entry>
  <entry>
    <string>fastq1</string>
    <de.uni__luebeck.inb.knowarc.usecases.ScriptInputUser>
      <tag>fastq1</tag>
      <file>false</file>
      <tempFile>false</tempFile>
      <binary>false</binary>
      <charsetName>windows-1252</charsetName>
      <forceCopy>false</forceCopy>
      <list>false</list>
      <concatenate>false</concatenate>
      <mime />
    </de.uni__luebeck.inb.knowarc.usecases.ScriptInputUser>
  </entry>
</inputs>
```

```
    <outputs />
    <includeStdIn>false</includeStdIn>
    <includeStdOut>true</includeStdOut>
    <includeStdErr>true</includeStdErr>
    <validReturnCodes>
      <int>0</int>
    </validReturnCodes>
  </useCaseDescription>
  <edited>false</edited>
</net.sf.taverna.t2.activities.externaltool.ExternalToolActivityConfigurationBean></configBean><annotations
/></activity></activities><dispatchStack><dispatchLayer><raven><group>net.sf.taverna.t2.core</group><artifact>workflowmodel-
impl</artifact><version>1.5</version></raven><class>net.sf.taverna.t2.workflowmodel.processor.dispatch.layers.Parallelize</class><configBean
encoding="xstream"><net.sf.taverna.t2.workflowmodel.processor.dispatch.layers.ParallelizeConfig xmlns="">
  <maxJobs>1</maxJobs>
</net.sf.taverna.t2.workflowmodel.processor.dispatch.layers.ParallelizeConfig></configBean></dispatchLayer><dispatchLayer><raven><group>net.sf.taverna.t2.core</group><artifact>workflowmodel-
impl</artifact><version>1.5</version></raven><class>net.sf.taverna.t2.workflowmodel.processor.dispatch.layers.ErrorBounce</class><configBean encoding="xstream"><null xmlns="" /></configBean></dispatchLayer><dispatchLayer><raven><group>net.sf.taverna.t2.core</group><artifact>workflowmodel-
impl</artifact><version>1.5</version></raven><class>net.sf.taverna.t2.workflowmodel.processor.dispatch.layers.Failover</class><configBean encoding="xstream"><null xmlns="" /></configBean></dispatchLayer><dispatchLayer><raven><group>net.sf.taverna.t2.core</group><artifact>workflowmodel-
impl</artifact><version>1.5</version></raven><class>net.sf.taverna.t2.workflowmodel.processor.dispatch.layers.Retry</class><configBean
encoding="xstream"><net.sf.taverna.t2.workflowmodel.processor.dispatch.layers.RetryConfig xmlns="">
  <backoffFactor>1.0</backoffFactor>
  <initialDelay>1000</initialDelay>
  <maxDelay>5000</maxDelay>
  <maxRetries>0</maxRetries>
</net.sf.taverna.t2.workflowmodel.processor.dispatch.layers.RetryConfig></configBean></dispatchLayer><dispatchLayer><raven><group>net.sf.taverna.t2.core</group><artifact>workflowmodel-
impl</artifact><version>1.5</version></raven><class>net.sf.taverna.t2.workflowmodel.processor.dispatch.layers.Invoke</class><configBean encoding="xstream"><null xmlns="" /></configBean></dispatchLayer></dispatchStack><iterationStrategyStack><iteration><strategy><cross><port name="input_dir" depth="0" /><port name="fastq1" depth="0" /></cross></strategy></iteration></iterationStrategyStack></processor></processors><conditions><condition control="ena_downloader" target="mapping" /><condition control="mapping" target="Deep_Variant" /></conditions><datalinks><datalink><sink type="processor"><processor>mapping</processor><port>ref_hgen</port></sink><source
```

type="dataflow"><port>ref_hgen</port></source></datalink><datalink><sink type="processor"><processor>mapping</processor><port>output_dir</port></sink><source type="dataflow"><port>output_dir</port></source></datalink><datalink><sink type="processor"><processor>mapping</processor><port>input_dir</port></sink><source type="dataflow"><port>input_dir</port></source></datalink><datalink><sink type="processor"><processor>mapping</processor><port>maptype</port></sink><source type="dataflow"><port>maptype</port></source></datalink><datalink><sink type="processor"><processor>mapping</processor><port>fastq1</port></sink><source type="dataflow"><port>fastq1</port></source></datalink><datalink><sink type="processor"><processor>mapping</processor><port>cpu</port></sink><source type="dataflow"><port>cpu</port></source></datalink><datalink><sink type="processor"><processor>Deep_Variant</processor><port>output_dir</port></sink><source type="dataflow"><port>output_dir</port></source></datalink><datalink><sink type="processor"><processor>Deep_Variant</processor><port>ref_hgen</port></sink><source type="dataflow"><port>ref_hgen</port></source></datalink><datalink><sink type="processor"><processor>Deep_Variant</processor><port>maptype</port></sink><source type="dataflow"><port>maptype</port></source></datalink><datalink><sink type="processor"><processor>Deep_Variant</processor><port>fastq1</port></sink><source type="dataflow"><port>fastq1</port></source></datalink><datalink><sink type="processor"><processor>Deep_Variant</processor><port>cpu</port></sink><source type="dataflow"><port>cpu</port></source></datalink><datalink><sink type="processor"><processor>ena_downloader</processor><port>input_dir</port></sink><source type="dataflow"><port>input_dir</port></source></datalink><datalink><sink type="processor"><processor>ena_downloader</processor><port>fastq1</port></sink><source type="dataflow"><port>fastq1</port></source></datalink><datalink><sink type="dataflow"><port>ena_downloader</port></sink><source type="processor"><processor>ena_downloader</processor><port>STDOUT</port></source></datalink><datalink><sink type="dataflow"><port>mapping</port></sink><source type="processor"><processor>mapping</processor><port>STDOUT</port></source></datalink><datalink><sink type="dataflow"><port>Deep_Variant</port></sink><source type="processor"><processor>Deep_Variant</processor><port>STDOUT</port></source></datalink></datalinks><annotations><annotation_chain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
      <annotationBean class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>bf582ab4-ce96-44b3-b1c4-d25138b7e43b</identification>
      </annotationBean>
      <date>2022-05-05 08:15:35.839 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_chain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>

```
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>e356ebdc-e832-4ec6-9fd6-b7149fe02867</identification>
      </annotationBean>
      <date>2022-05-05 16:35:57.595 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_c
hain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>9a8f67cc-5f30-4932-9610-e7f1c5e5316e</identification>
      </annotationBean>
      <date>2022-05-04 14:19:00.638 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_c
hain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>286c81e7-a774-450c-90de-28dc7ee60d98</identification>
      </annotationBean>
      <date>2022-05-07 10:42:20.691 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_c
hain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
```

```
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>e26ce0b3-36ec-4352-8c2a-b42ee8586860</identification>
      </annotationBean>
      <date>2022-05-05 16:36:37.275 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_c
hain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>c9f912b4-524a-4916-acac-526f655e1220</identification>
      </annotationBean>
      <date>2022-05-05 07:22:24.37 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_c
hain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>fcb40c2f-1e7f-4c20-b672-ccf24deb1975</identification>
      </annotationBean>
      <date>2022-05-07 07:47:46.252 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_c
hain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
```

```
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>d368b641-1d9f-4f51-bc59-63288743e558</identification>
      </annotationBean>
      <date>2022-05-07 07:48:58.935 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_c
hain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>b26abfc6-96f1-4a8d-b12d-ebe062227e68</identification>
      </annotationBean>
      <date>2022-05-05 06:59:25.480 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_c
hain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>2afa277d-addc-4be5-9e6f-554cc7796c53</identification>
      </annotationBean>
      <date>2022-05-07 07:47:33.332 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_c
hain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
```

```
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>cd680466-addb-4c1b-9b6a-f603894190d6</identification>
      </annotationBean>
      <date>2022-05-04 11:51:14.170 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_c
hain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>c7c00272-4f27-45b7-95eb-b38d04a8662e</identification>
      </annotationBean>
      <date>2022-05-05 08:26:38.907 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_c
hain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>2830118d-9a9d-4380-a82e-81ed6be1cff6</identification>
      </annotationBean>
      <date>2022-05-06 10:57:41.391 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_c
hain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
```

<annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
<identification>909ac05e-1ab0-4a66-846b-d0e6e860a1ec</identification>
</annotationBean>
<date>2022-05-04 14:14:52.198 UTC</date>
<creators />
<curationEventList />
</net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
</annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_chain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
<annotationAssertions>
<net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
<annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
<identification>186d555f-aa7f-46c6-9a07-8eda31cbb16e</identification>
</annotationBean>
<date>2022-05-04 13:32:52.728 UTC</date>
<creators />
<curationEventList />
</net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
</annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_chain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
<annotationAssertions>
<net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
<annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
<identification>ac7d142e-0db2-4992-aaee-475884f59ec7</identification>
</annotationBean>
<date>2022-05-06 07:39:31.388 UTC</date>
<creators />
<curationEventList />
</net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
</annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_chain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
<annotationAssertions>
<net.sf.taverna.t2.annotation.AnnotationAssertionImpl>

<annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
<identification>a5b5eb6a-62ed-48a4-a802-c1de5c47d523</identification>
</annotationBean>
<date>2022-05-07 09:15:41.445 UTC</date>
<creators />
<curationEventList />
</net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
</annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_chain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
<annotationAssertions>
<net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
<annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
<identification>64c43d80-0373-492a-acec-7701978d178d</identification>
</annotationBean>
<date>2022-05-05 06:48:14.230 UTC</date>
<creators />
<curationEventList />
</net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
</annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_chain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
<annotationAssertions>
<net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
<annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
<identification>55100d64-5740-4816-ac0a-243f0b2e96d7</identification>
</annotationBean>
<date>2022-05-05 08:06:58.54 UTC</date>
<creators />
<curationEventList />
</net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
</annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_chain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
<annotationAssertions>
<net.sf.taverna.t2.annotation.AnnotationAssertionImpl>

```
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>bacc2a7e-e0d5-499a-b55d-67007841a972</identification>
      </annotationBean>
      <date>2022-05-07 10:05:08.887 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_c
hain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>409a5cbc-459d-43a7-a667-b4e5c23457d6</identification>
      </annotationBean>
      <date>2022-05-23 09:53:28.251 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_c
hain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>bb40a405-e2c8-4b76-904e-90fb5786b68b</identification>
      </annotationBean>
      <date>2022-05-23 09:56:10.21 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_c
hain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
```

```
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>fa90ad83-29f4-4967-aab3-49bc40cffbbc</identification>
      </annotationBean>
      <date>2022-05-06 05:50:12.824 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_c
hain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>28255bca-da42-45ce-87ed-81fa90c33568</identification>
      </annotationBean>
      <date>2022-05-23 10:08:16.704 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_c
hain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>4c2dd107-972e-49fc-b56b-146981940a51</identification>
      </annotationBean>
      <date>2022-05-12 14:23:08.770 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_c
hain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
```

```
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>cd8159fa-62cb-445b-bdd2-a22dea00b000</identification>
      </annotationBean>
      <date>2022-05-07 07:54:59.523 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_c
hain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>8e3f9f1b-4a20-4ab0-ad1a-7eedf996d2f7</identification>
      </annotationBean>
      <date>2022-05-05 16:53:13.146 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_c
hain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>26a5b42b-025b-41b1-af24-722406742dc6</identification>
      </annotationBean>
      <date>2022-05-12 14:14:40.151 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_c
hain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
```

```
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>e44d9825-ddb7-4b7b-867f-4347a2db980f</identification>
      </annotationBean>
      <date>2022-05-05 07:50:26.861 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_c
hain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>9f06f89d-8ce7-41a7-87d4-2c2eec8dc273</identification>
      </annotationBean>
      <date>2022-05-04 12:05:58.561 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_c
hain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>e1912994-8e37-4750-a25e-5ce60b8ff1dc</identification>
      </annotationBean>
      <date>2022-05-05 19:43:10.696 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_c
hain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
```

```
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>a1e8b76b-583d-4661-8e0a-50b4ec2442c5</identification>
      </annotationBean>
      <date>2022-05-04 13:59:23.714 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_c
hain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>789d879f-99f2-41bc-a86b-7ad1404d213f</identification>
      </annotationBean>
      <date>2022-05-04 14:16:30.987 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_c
hain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>10cf4b7b-3d2b-4bd2-899b-7ba11316b11b</identification>
      </annotationBean>
      <date>2022-05-12 14:06:37.641 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_c
hain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
```

```
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>d7532bf9-51ef-4d91-aaa5-b5b93ac38668</identification>
      </annotationBean>
      <date>2022-05-07 10:02:31.708 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_c
hain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>048c9ea7-6b6d-4715-8454-a0d191649017</identification>
      </annotationBean>
      <date>2022-05-05 06:30:24.283 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_c
hain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>48a275c5-92ee-4c60-8ed8-1cc7a4256779</identification>
      </annotationBean>
      <date>2022-05-05 07:45:50.423 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_c
hain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
```

```
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>56a3c154-e6e1-4787-ab2a-d1f5865207fa</identification>
      </annotationBean>
      <date>2022-05-04 11:50:33.71 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_c
hain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>94e43334-d62d-41be-898d-24838d29f6d1</identification>
      </annotationBean>
      <date>2022-05-04 14:59:44.793 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_c
hain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>f019ebe1-8b18-4036-8ad6-87ed1a707ec2</identification>
      </annotationBean>
      <date>2022-05-04 15:01:59.219 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_c
hain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
```

```
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>41fd26d1-65fc-47a1-ab21-6fb1f31fdaa6</identification>
      </annotationBean>
      <date>2022-05-07 09:54:54.263 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_c
hain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>8fbc405d-d1eb-4057-a7c3-af82caad1f76</identification>
      </annotationBean>
      <date>2022-05-12 14:16:11.430 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_c
hain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>78ec372b-d38b-451c-b36d-18bd78057ecf</identification>
      </annotationBean>
      <date>2022-05-07 09:48:20.216 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_c
hain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
```

```
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>b8fca280-11e9-432c-abc0-57eac89c00ef</identification>
      </annotationBean>
      <date>2022-05-12 13:59:39.649 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2><annotation_c
hain_2_2 encoding="xstream"><net.sf.taverna.t2.annotation.AnnotationChainImpl xmlns="">
  <annotationAssertions>
    <net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
      <annotationBean
class="net.sf.taverna.t2.annotation.annotationbeans.IdentificationAssertion">
        <identification>2da50e5a-bfc3-43cf-a1bc-e5444d84903b</identification>
      </annotationBean>
      <date>2022-05-04 14:15:37.315 UTC</date>
      <creators />
      <curationEventList />
    </net.sf.taverna.t2.annotation.AnnotationAssertionImpl>
  </annotationAssertions>
</net.sf.taverna.t2.annotation.AnnotationChainImpl></annotation_chain_2_2></annotations
></dataflow></workflow>
```