Quantificació de mostres d'ADN amb mètodes computacionalment eficients

Esteve Pineda Sánchez

Resum– Avui dia existeixen multitud de tècniques de seqüenciació d'ADN i mètodes d'estudi per poder analitzar mostres complexes. El projecte desenvolupat al llarg dels últims mesos consisteix en treballar amb diferents algorismes i conjunts de dades (parell de bases d'ADN d'insectes agrupades en arxius de 150 bp aproximadament) extretes per poder dur a terme una quantificació de mostres d'ADN utilitzant mètodes computacionalment eficients.
Aquesta proposta s'ha orientat al disseny i avaluació de diferents estratègies que permetin classificar de forma quantitativa les diferents espècies que formen part d'una mostra. Per poder realitzar aquesta tasca s'ha partit de solucions desenvolupades en el departament d'Arquitectura de Computadors i Sistemes Operatius per tal de generar versions millorades i poder realitzar un estudi sistemàtic. En aquest article, s'hi troba sintetitzada i ordenada tota la feina realitzada durant els últims mesos.

Paraules clau– Metagenòmica, Kraken2, BWA, ADN, SLURM, anàlisi de rendiment, Bracken, seqüències de comandes bash, Python, Clúster.

Abstract– Nowadays, there is a multitude of DNA sequencing techniques and study methods available to analyse complex samples. The project developed over the last few months consisted of working with different algorithms and datasets (base pair of insect DNA bases grouped in files of approximately 150 bp) extracted to be able to perform a quantification of DNA samples using computationally efficient methods.
This proposal has been focused on the design and evaluation of different strategies that allow to quantitative classify the different species that take part of a sample. To carry out this task, solutions developed in the Computer Architecture & Operating Systems Department have been used as a starting point, in order to generate improved versions that allow a final systematic study to be conducted. This article summarises and organises all the work carried out over the last months.

Keywords– Metagenomics, Kraken2, BWA, DNA, SLURM, performance analisys, Bracken, bash scripting, Python, Cluster.

1 Introducció

En bioinformàtica existeix una tècnica de seqüenciació d'alt rendiment bastant novedosa que s'utilitza per identificar i quantificar el número d'espècies presents en una mostra, permetent arribar a identificar organismes que amb altres mètodes passarien desapercebuts.

Aquesta tècnica en qüestió, pertany a l'àmbit de la metagenòmica i consisteix en utilitzar el que es coneix com a "shotgun metagenomics" per intentar tallar una mostra d'ADN en molts trossos més petits (lectures en brut agrupades en parells de bases). Amb aquestes noves mostres de tamany reduït, fent ús d'algorismes que es descriuen al llarg d'aquest article, es pot determinar a quina espècie pertany un organisme identificat en cada lectura (o "read"). Finalment, utilitzant tota la informació generada per aquests algorismes, es pot obtenir l'abundància d'una espècie dins del conjunt de la mostra inicial.

En aquest treball de final de grau s'ha treballat amb diferents algorismes, implementats dins d'un pipeline, i que alhora, utilitzen diferents conjunts de dades. Aquestes dades es troben organitzades en parell de bases d'ADN d'insectes agrupades en arxius de 150 pb aproximadament, i que s'han extret per intentar realitzar una quantificació de mostres d'ADN utilitzant mètodes computacionalment eficients.

Aquest article es troba organitzat en diferents seccions i subseccions dedicades a contenir una determinada fracció del treball global per tal de donar un fil conductor a l'article

• E-mail de contacte: Esteve.Pineda@autonoma.cat
• Menció realitzada: Enginyeria de Computadors
• Treball tutoritzat per: Miquel Àngel Senar (DACSO)
• Curs 2021/22

i sintetitzar la informació de forma ordenada:

- Una primera part introductòria en la que s'ha descrit el context del treball, les motivacions i els objectius principals del treball.
- A continuació s'hi troba descrit breument l'estat de l'art en el que s'hi descriuen estudis i eines relacionades amb aquest treball.
- Tot seguit es descriu una mica quina metodologia de treball i quina planificació s'ha seguit al llarg dels últims mesos per poder realitzar aquest estudi.
- Més endavant, es troba definida quina ha estat la meva aportació pròpia als estudis inicials i de quina manera s'han dissenyat i implementat els diferents pipelines per realitzar una classificació de les diferents espècies existents en una mostra d'ADN.
- La part final d'aquest article, està centrada en la fase d'anàlisi realitzada. Quins estudis s'han realitzat, quins resultats s'han obtingut, i quines conclusions se'n poden extreure dels mateixos.
- I finalment, s'ha definit una bibliografia amb la documentació revisada més important.

1.1 Objectius

La idea principal d'aquest projecte no es basa tant en intentar paral·lelitzar seccions de codi per intentar reduir el seu temps de còmput utilitzant llenguatges com OpenCL, CUDA, MPI... sino que va més enfocada en implementar algorismes d'optimització que permetin eliminar càlculs o etapes innecessàries i estudiar el seu comportament dins d'un pipeline per poder prendre diferents decisions (per exemple permetre petits marges d'error per tal d'aprofitar una reducció en el temps d'execució, entre d'altres aspectes...).

De cara a establir un punt de partida del treball a realitzar durant aquest període de temps es van proposar una sèrie d'objectius més específics a assolir i d'aquesta manera tenir uns objectius clars i unes pautes més marcades:

- Adquirir una metodologia de treball adequada per poder adquirir la major quantitat de coneixements.
- Entendre el funcionament de noves eines de classificació de mostres d'ADN i de quina manera poden interactuar entre elles.
- Trobar mètodes de classificació de mostres d'ADN més eficients i ràpids (respectant el nivell de fiabilitat en les seves classificacions).
- Adquirir experiència en elaborar un treball a partir d'un estudi / investigació dut a terme per una altra persona per poder-lo completar minimament.
- Aprendre o fer un recordatori sobre l'ús de diferents llenguatges de programació (Bash, Python, C...) així com aprendre a utilitzar eines de gestió de clúster (SLURM...).
- Ser capaç de sintetitzar tota la feina realitzada de forma ordenada i entenedora en un article estandaritzat.

2 ESTAT DE L'ART

El punt de partida d'aquest treball, consisteix en l'anàlisi d'un pipeline implementat per la Lidia Garrido (proporcionat pel departament DACSO i que es descriu de forma més completa a l'article [1]) per tal d'identificar els diferents elements que intervenen.

El pipeline en qüestió (amb el que es compararan les noves versions per determinar si s'ha pogut millorar l'eficiència inicial o no i que s'observa a la figura 1) està format per dues etapes principals, una primera etapa de mapeig i una segona etapa d'assignació.

Per poder fer servir aquests algoritmes, cal una sèrie de dades d'entrada. Per una banda es necessita la mostra d'ADN a estudiar (seqüenciada en N reads de 150 bp aproximadament) i per una altra banda es necessiten els genomes de referència amb els que poder comparar cadascun dels reads d'entrada.

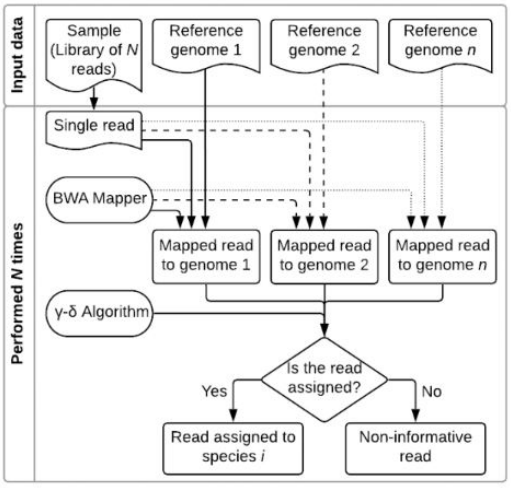

Fig. 1: Diagrama de flux del pipeline original

Etapa de mapeig

El primer algorisme que intervé un cop preparades les dades d'entrada, és un algorisme conegut com a BWA Mapper[2] i que s'encarrega de realitzar la fase d'alineament que consisteix en realitzar el mapeig d'un read per a cada genoma de referència generant així un arxiu d'alineació per referència.

Aquesta és l'etapa del pipeline anterior més costosa computacionalment parlant, ja que l'algorisme BWA consumeix aproximadament el 89% del temps total que triga el pipeline i que segons un estudi previ, utilitzant 6 cores amb 24 threads per core el temps pot variar entre 54 min (1,3 milions de reads) i 1h 16 min (1,9 milions de reads) en funció del número de reads a tractar.

Etapa d'assignació

Pot donar-se el cas en que un read coincideixi amb part de més d'un genoma de referència i per aquest motiu es necessita un algorisme que decideixi a quin genoma i, per tant, espècie asignar aquell read o si pel contrari no hi ha evidències suficients per a determinar això i per tant descartar-lo. Aquest algorisme es coneix com algorisme $\gamma - \delta$.

Inicialment s'agafen les dues proporcions de mapeig més altes (obtingudes en dividir el nombre de nucleòtids coincidents amb un dels genomes de referència amb el nombre de nucleòtids total) i es comparen amb els paràmetres γ i δ que s'han definit prèviament utilitzant un set d'entrenament amb un 75% de mostres triades aleatòriament i un 25% restant per a realitzar una sèrie de tests.

En funció del resultat d'aquesta comparació, l'algorisme assigna aquell read a una espècie concreta o el descarta per falta de determinisme.

És important tenir en compte que per a que aquest algorisme es comporti com cal, γ ha de ser sempre superior a δ.

A la figura següent (figura 2) es troba el diagrama de flux de l'algorisme γ–δ descrit als paràgrafs anteriors.

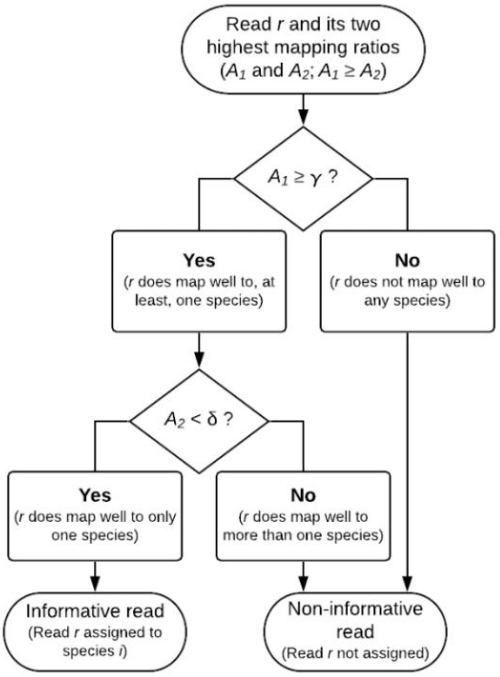

Fig. 2: Diagrama de flux de l'algorisme $\gamma - \delta$

3 METODOLOGIA I PLANIFICACIÓ

Per poder treballar de forma ordenada i poder abordar els diferents temes i objectius a assolir, una de les primeres coses que es va fer va ser establir una metodologia de treball adequada per dur a terme un bon treball de final de grau.

Durant els diferents períodes entre entregues parcials, s'ha seguit la metodologia següent:

La idea principal que defineix una mica la metodologia seguida durant aquest període de temps ha consistit en anar progressant poc a poc, intentant dividir els problemes "grans" en altres problemes més petits aplicant així la tècnica coneguda com a "Divide and Conquer".

Inicialment és important realitzar una tasca de documentació (sempre que sigui necessària) per poder fer un estudi sobre quina és la millor manera d'afrontar un problema concret en cada cas.

Seguidament és interessant planificar minimament quin ha de ser el pla de treball a seguir per centrar-se en la qüestió important a tractar en cada moment i deixar de banda aspectes que en aquell instant no siguin rellevants i d'aquesta manera dedicar els esforços en resoldre un problema concret per poder avançar amb el treball, ja sigui implementant una hipòtesi inicial o bé descartant-la.

Un cop implementada una nova funcionalitat o fet un canvi en el pipeline inicial, és important analitzar els resultats obtinguts a causa d'això i poder decidir si el canvi realment ha permès millorar les prestacions de l'algorisme inicial o no i en aquest últim cas, entendre per quin motiu no ha millorat el temps d'execució previ.

Finalment, s'ha de deixar documentat en un informe tota la feina que s'hagi estat realitzant a l'hora d'implementar els canvis anteriors, per una banda per deixar constància de la feina que s'ha anat realitzant i per una altra banda per facilitar la síntesi del treball a l'informe final.

En quant a quina ha estat la planificació definida per dur a terme aquest treball, és interessant comentar que s'ha dividit la planificació en períodes de setmanes / mesos entre les entregues parcials que s'han hagut de realitzar durant aquest temps.

Durant les primeres setmanes, la feina a realitzar va consistir en documentar-se sobre tot el rerefons que hi ha respecte a la temàtica del treball, entendre el funcionament del pipeline original i començar a treballar amb les eines necessàries (gestor de cues SLURM, editor de linux vim, sistemes de fitxers compartits...).

De cara al segon període del treball, la feina s'ha centrat en aprendre el funcionament i implementar una nova eina (que es defineix i s'explica a la secció 4, Kraken2) sobre el pipeline original i verificar el correcte funcionament del pipeline.

Finalment en el període de temps restant el treball a realitzar ha consistit en implementar un nou pipeline per tal d'utilitzar una nova eina anomenada Bracken (que utilitza per sota Kraken2) i que permet obtenir l'abundància d'espècies que existeix en una mostra i realitzar un estudi sistemàtic sobre l'eficiència que ofereixen aquests nous pipelines (rendiment en funció del temps d'execució, número de mostres classificades...).

Un cop finalitzada tota la feina, s'ha hagut de realitzar un informe de caràcter tècnic per explicar una mica quin ha estat el resultat del projecte (a part d'explicar també les diferents fases que l'han conformat).

4 APORTACIÓ PRÒPIA

Tal i com s'ha comentat en la introducció d'aquest article, l'objectiu principal és el d'utilitzar mètodes de classificació eficients intentant reduir en tant que es pugui la realització d'etapes i/o càlculs innecessaris.

En aquest punt és on entren els algorismes que poden ser anomenats "Lowest Common Ancestor" i que enlloc de descartar un read tal i com fa l'algorisme $\gamma - \delta$ per falta de determinisme a l'assignar una espècie o una altra, el que es fa és assignar la mostra al seu ancestre comú més proper dins de l'arbre taxonòmic i per tant ens pot permetre eliminar tots aquells reads/mostres d'entrada que no hagin estat assignades a una espècie en concret (o un altre nivell taxonòmic que podem triar) però sí que hagin estat assignades a un ancestre comú.

La primera opció que ens ha permès introduir aquest canvi ha estat l'ús de la segona versió de l'alineador conegut

com a "Kraken"[4].

El motiu principal pel qual es pretén realitzar aquesta implementació és deu principalment a la rapidesa d'aquests algorismes per fer aquests tipus de classificació i per tant ajudar a reduir el temps d'execució de la resta del pipeline degut a la reducció de mostres inicials a comparar (principalment s'espera aconseguir una reducció del temps d'execució en l'etapa d'alineament inicial per part de l'algorisme BWA).

Si més no, Kraken és un alineador que no calcula l'abundància d'una espècie dins d'una mostra, per aquest motiu, i per poder estalviar-se l'ús dels algorismes originals BWA i $\gamma - \delta$ (i el temps que triguen en executar-se), s'ha proposat l'ús d'una eina de classificació diferent que permet calcular l'abundància d'un cert nivell taxonòmic (ordre, família, espècie...) de manera molt ràpida (tot i que amb molta probabilitat, el número de reads classificats finalment seran inferiors als reads classificats amb la resta d'algorismes), Bracken[5].

Kraken2

Kraken2, com ja s'ha comentat prèviament, és un alineador que permet classificar mostres dins d'un arbre taxonòmic determinat de forma ràpida i precisa.

El principi de funcionament d'aquesta eina es basa en fer la coincidència dels diferents k-mers que conté una mostra d'entrada, dins d'una seqüència de l'ancestre comú més baix dins d'un arbre taxonòmic. Per poder fer aquesta comparació i comprovar a quin nivell (si és que hi ha informació suficient a la mostra) de l'arbre taxonòmic pertany l'insecte seqüenciat en una mostra, Kraken2 utilitza una taula hash generada a partir dels genomes de referència introduïts en el moment de la generació d'aquesta base de dades.

El comportament de Kraken2 es pot veure compromès si en el moment de generar la base de dades (i per tant la taula hash) es decideix establir un tamany màxim inferior al que necessitaria normalment. El motiu principal pel qual es pot arribar a prendre aquesta decisió pot ser degut al poc espai d'emmagatzematge de la màquina utilitzada.

A l'hora de generar una nova base de dades de Kraken2, hi ha diferents aspectes molt important a considerar:

- S'ha de tenir present de quanta memòria RAM disposa la màquina a utilitzar, ja que en el moment en que s'utilitza kraken, la taula hash necessita poder-se carregar a memòria.
- S'ha de tenir espai suficient en disc per poder generar la base de dades (hi ha base de dades que poden arribar a necessitar 100GB en disc).
- Tot i que l'alineador Kraken2 és molt ràpid i eficient, la generació de la Base de Dades no ho és tant i en funció del tamany d'aquesta, el temps necessari requerit pot ser força elevat.

A continuació es descriu el funcionament del pipeline de Kraken que s'ha implementat prèviament a l'algorisme BWA del pipeline original per intentar reduir el número de mostres d'entrada i per tant, intentar reduir el temps d'execució global (figura 3).

Tal i com s'observa a la figura 3, Kraken2 només necessita dos paràmetres (obligatoris, després es poden modificar certs paràmetres com per exemple l'índex de confiança, però en aquest cas no ha estat necessari), per una banda la llibreria amb els reads d'entrada i el path on es troba la base de dades generades.

Kraken2 genera dos fitxers addicionals, un report amb informació resumida sobre els reads classificats, i un fitxer de sortida amb extensió .kraken que guarda una línea amb cinc camps amb informació rellevant sobre el que ha succeït amb cada mostra en particular.

Un cop l'alineador ha classificat les mostres d'entrada i s'han generat els dos fitxers anteriors, s'ha utilitzat una eina compatible amb kraken (extract_kraken_reads.py[8]) que permet generar un nou fitxer amb extensió .fastq amb totes les mostres que s'hagin classificat a partir d'un cert nivell taxonòmic especificat (fitxer que s'utilitzarà com a dades d'entrada de l'algorisme BWA).

És important tenir present, que al llarg dels estudis realitzats durant aquest període, s'han filtrat totes les mostres a partir del nivell taxonòmic "familía".

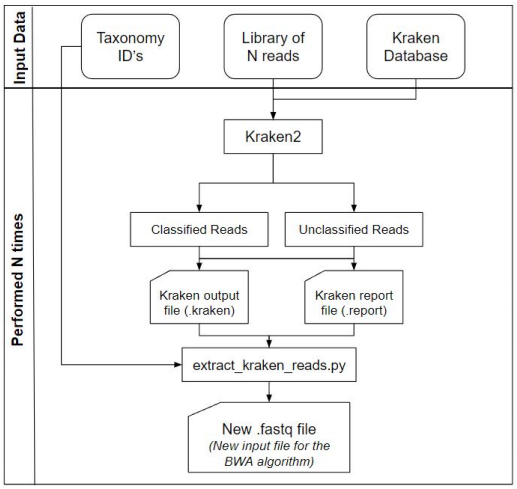

Fig. 3: Diagrama de flux del pipeline implementat per utilitzar Kraken2

Bracken

L'última eina utilitzada durant aquest treball ha estat Bracken, i en conjunt amb l'alineador Kraken2, ha servit per comparar l'ús d'eines més lentes però més precises amb eines més ràpides però menys precises i alhora poder obtenir l'abundancia de les diferents espècies que puguin trobar-se en una mostra.

Abans de poder utilitzar Bracken però, s'ha de generar un fitxer de base de dades de Bracken especificant els següents paràmentres:

- Path de la base de dades de Kraken2
- Número de threads amb els que generar aquest fitxer.
- Longitud dels k-mers utilitzats en la base de dades de Kraken2.
- Tamany de les mostres d'entrada (en parell de bases)

La mida d'aquest fitxer i el temps d'execució necessari pot variar en funció de la mida de la base de dades de Kraken2

A continuació (figura 4) es descriu el diagrama de flux que segueix una execució habitual de Bracken.

Per començar, s'ha de realitzar una primera classificació amb Kraken2 tal i com s'ha descrit a l'apartat anterior (refernt a Kraken2). En aquest punt, enlloc d'utilitzar l'eina extract_kraken_reads.py s'ha utilitzat Bracken per obtenir directament l'abundància d'una espècie (o un altre nivell taxonòmic) en una mostra.

Per tant, per poder utilitzar Bracken, es necessiten uns paràmetres addicionals:

- Path de la base de dades de Kraken2
- Nivell taxonòmic a partir del qual realitzar el càlcul de l'abundància en la mostra d'entrada.
- Threshold per indicar el número de mostres mínimes que ha de tenir l'element taxonòmic per poder considerarlo a l'hora de realitzar el càlcul de l'abundància.
- Tamany de les mostres d'entrada (en parell de bases)

Bracken finalment genera un nou fitxer .bracken amb tota la nova informació referent a l'abundància d'una espècie en una mostra d'entrada.

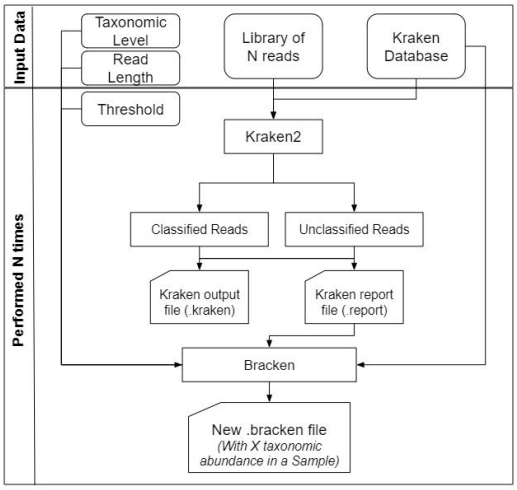

Fig. 4: Diagrama de flux del pipeline implementat per utilitzar Bracken

5 FASE D'ANÀLISI

L'última secció d'aquest informe, està dedicat a la realització d'un estudi sobre diferents aspectes relacionats amb el comportament dels diferents pipelines implementats, així com dels resultats que se'n deriven en cada cas. Abans de comentar els resultats obtinguts és important definir / identificar l'entorn en el qual s'han produit els experiments i d'aquesta manera poder justificar totes les decisions que s'hagin arribat a prendre.

5.1 Consideracions prèvies

Al llarg d'aquest període de temps, s'ha treballat principalment amb els diferents nodes que pertanyen a la partició nodo.q del clúster del que disposa la Universitat Autònoma de Barcelona "Wilma".

Les característiques més importants a considerar de cara poder executar els diferents pipelines, en aquest cas han estat:

- Espai disponible en disc.
- Quantitat de memòria RAM de la que disposa cada node.
- Quina quantitat de threads es poden utilitzar en cada màquina (i per tant també és important tenir present el número de cpu's de les que disposa cada node).

Dels anteriors elements, es pot considerar la memòria RAM com l'element més important de cara a l'ús de Kraken2, ja que com s'ha comentat prèviament (secció 4), Kraken2 necessita poder carregar la taula hash en memòria, i per tant el tamany d'aquesta taula (i alhora, els resultats obtinguts) dependran del tamany de la RAM disponible en el node en qüestió.

5.2 Estudi sistemàtic realitzat

La idea inicial per tal de realitzar un estudi sistemàtic, era la d'utilitzar per una banda el pipeline original i per una altra el pipeline implementat amb Kraken2 i tractar de comparar-los a partir de les seves execucions fent servir llibreries amb les mostres tant de Single-Species com de Mixed-Species. I finalment utilitzar el pipeline en el que s'implementa Bracken per poder extreure una sèrie de conclusions (que s'acaben describint al llarg d'aquesta secció).

Abans d'executar cap pipeline, s'ha de determinar la capacitat dels nodes als que s'enviaran els diferents treballs, ja que com ja s'ha comentat en l'apartat anterior 5.1, el comportament de Kraken2 variarà en funció de la capacitat de la memòria RAM. Per una altra banda, s'han de considerar també la quantitat de threads màxims que es poden utilitzar en aquests nodes, ja que això permet definir el nivell de paral·lelisme amb el que poden treballar els altres dos algorismes (per una banda l'alineador BWA i per una altra banda l'algorisme $\gamma - \delta$).

En aquest cas, els nodes de la partició disponible del clúster Wilma amb el que em pogut treballar (partició nodo.q) disposen d'una memòria RAM de 8GB i de 12 cores que poden executar 1 thread/core. Amb aquests valors, després d'investigar una mica més quin era el tamany òptim que necessitava Kraken2 per funcionar amb la millor fiabilitat i completitud possible, ens adonem de que Kraken2 normalment necessita +45GB aproximadament de memòria RAM (tamany de la taula hash, tot i que en espai en disc aquest valor s'incrementa per poder guardar altres dades necessàries a l'hora de generar aquesta base de dades).

Per aquest motiu, es decideix generar dos pipelines basats en la utilització de Kraken2 diferents, de manera que un pugui ser executat completament en els nodes més senzills (partició nodo.q), generant així una taula hash de 5GB (considerant que només tenim 8GB de memòria RAM) mentre que el segon pipeline, executi la part de Kraken2 en una partició diferent (partició research.q, molt més potent, que utilitzen els alumnes de màster) disposant així d'una memòria RAM de 64GB que ens ha permès generar una taula hash de 50GB.

A mode de síntesi, finalment s'han executat tres pipelines diferents (original, Kraken 5G + BWA + $\gamma - \delta$ i finalment Kraken 50G + BWA + $\gamma - \delta$) per poder extreure una sèrie de conclusions relacionades amb el seu comportament (número de mostres assignades, temps d'execucions...).

Per una altra banda, s'han realitzat dos experiments addicionals que han consistit en variar el número de mostres inicials que s'introdueixen al pipeline original per comprovar com afecta aquest canvi al temps d'execució final.

I finalment, s'ha realitzat l'execució del pipeline de Bracken per poder comparar l'abundància d'espècies obtinguda en cada cas (i poder determinar quin algorisme és més o menys fiable).

5.3 Comparativa i Resultats

Un cop realitzades totes les execucions necessàries per poder extreure els resultats pertinents s'han generat una sèrie de gràfics i taules per visualitzar-los de forma més visual i senzilla a l'hora de fer qualsevol tipus d'anàlisi.

Tant la primera figura (figura 5) com les taules 1, 3 i 4 (aquestes dues últimes afegides a l'apèndix degut al tamany que requereixen) contenen dades relacionades amb el número de reads assignats en els diferents pipelines, tant per a les llibreries Single-Species, com per a les llibreries Mixed-Species.

Si ens hi fixem a la figura 5 es pot observar de quina manera afecta el fet d'utilitzar una base de dades de Kraken més o menys gran. Tal i com s'observa en aquesta figura, és interessant poder utilitzar una base de dades quant més completa millor, tot i que això impliqui haver de requerir un equip de qualitat superior.

Tot i que el número de reads classificats final pugui semblar que no varia gaire (un 5,4% en el cas de les Single-Species i un 6,53% en el cas de les Mixed-Species), fixant-nos en les taules amb els valors absoluts, es pot comprovar com realment no son poques les mostres de diferència, les que s'arriben a assignar finalment per l'algorisme $\gamma - \delta$ en cada cas.

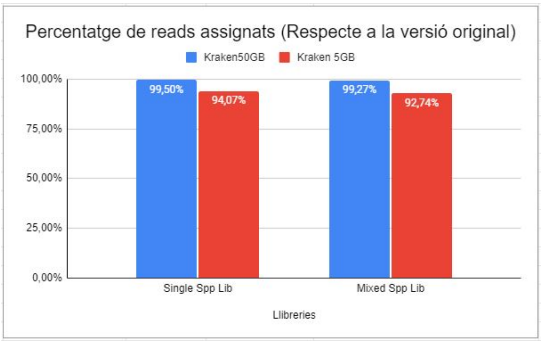

Fig. 5: Número de reads assignats [%]

TAULA 1: DIFERÈNCIA DE READS ASSIGNATS PER L'ALGORISME $\gamma - \delta$ RESPECTE DE LA VERSIÓ ORIGINAL

Diferència de reads assignats		
LLibreries	**Kraken 5GB**	**Kraken 50GB**
SingleSpp	667442	55815
MixedSpp	388152	38981

A continuació, a les dues figures següents (figura 6 i figura 7) s'hi troben els temps d'execució dels diferents pipelines per als dos tipus de llibreries.

Tal i com es podia esperar inicialment, el temps d'execució dels pipelines varia en funció del nombre de mostres d'entrades (sortida de l'alineador kraken) amb les que ha de treballar l'algorisme BWA. Per aquest motiu, el pipeline que treballa amb Kraken i utilitza una base de dades de 5GB triga menys temps en executar-se que la resta de pipelines (recordem que com que la Base de dades de 5GB és més reduïda, el número de mostres classificades per taxó per aquest alineador és reduït respecte la versió que utilitza la base de dades de 50GB).

S'ha de considerar, que les llibreries Mixed Species tenen un temps d'execució inferior (20,41h, 17,42h i 20h) respecte al temps obtinguts en executar els pipelines amb les llibreries Single-Species (46,93h, 43,02h i 44,05h) degut a la diferència de reads inicials totals que existeix.

És important comentar, que els temps obtinguts per al pipeline que fa servir la base de dades de 50GB son una mica orientatius i poden variar lleugerament. Això es deu principalment a que l'accés a les màquines de la partició research.q és limitat i per tant el temps corresponent a la part de l'execució de kraken2 s'ha obtingut en fer una estimació considerant el temps que triga kraken2 utilitzant la base de dades de 5GB i en funció del número de mostres d'entrada.

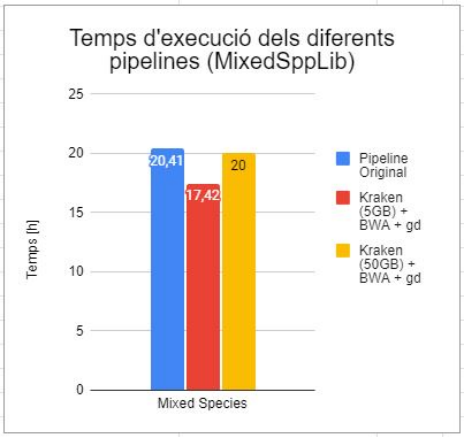

Fig. 6: Temps d'execució (en hores) dels diferents pipelines (MixedSppLib)

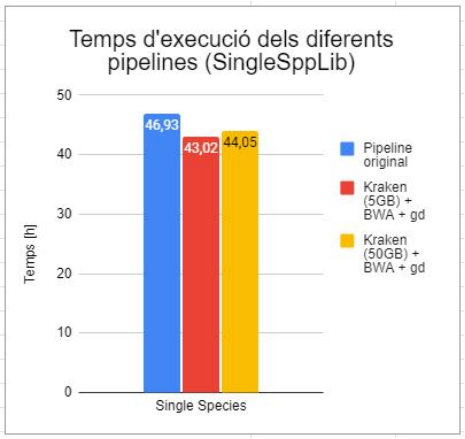

Fig. 7: Temps d'execució (en hores) dels diferents pipelines (SingleSppLib)

Un dels últims experiments realitzats ha consistit en variar el número de mostres d'entrada de l'algorisme BWA (utilitzant en aquest cas el pipeline original) per tal de comprovar si el temps d'execució augmenta linealment a mesura que s'incrementa el número de mostres d'entrada o no.

És important deixar constància de que el temps d'execució del pipeline per a una única mostra pot variar en funció del que trigui l'algorisme $\gamma - \delta$ en assignar-la a una espècie en concret. Això es deu principalment a que la mostra en un primer moment podria pertànyer a més d'una espècie i per tant l'algorisme ha de continuar aplicant certs criteris per veure si hi realment pot ser classificada en una única espècie o si realment no la pot classificar per falta de determinisme.

A la figura següent (figura 8) es pot observar com el temps d'execució augmenta linealment (tret de petites pertorbacions degut al que s'ha comentat en el paràgraf anterior) a mesura que el número de mostres d'entrada creix.

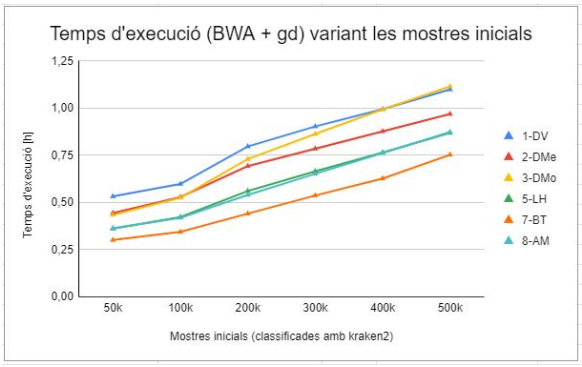

Fig. 8: Temps d'execució del pipeline original variant les mostres inicials

Finalment i per acabar amb la part d'experimentació, s'han implementat una sèrie d'scripts (que es poden trobar al repositori de GitHub del projecte [11]) per tal de poder extreure l'abundància de les diferents espècies un cop s'han classificat un cop executats el diferents pipelines. Per a aquest experiment, també s'ha executat el pipeline de Bracken per poder-lo comparar (en quant a temps i fiabilitat) amb la resta de pipelines.

Si ens hi fixem a la taula 2, es poden observar les espècies més abundants dintre de les llibreries de Mixed-Species un cop executats els diferents pipelines. És important tenir present que s'ha pres com a referència les abundàncies obtingudes en executar el pipeline original per tal de comparar de quina manera afecta l'ús de kraken i bracken respecte a la versió original. I finalment, s'ha de comentar que les espècies que apareixen com a referència (pipeline original) son espècies que han superat el límit de detecció establert, $\varepsilon = 0.01$.

Un aspecte important a considerar és el fet de que Bracken és una eina que encara està en constant desenvolupament i que en alguns casos es poden arribar a produir alguns bugs, per tant encara és una eina poc fiable per intentar calcular l'abundància d'espècies en una mostra.

Observant la taula 2, a grans trets es pot observar que tret d'alguns casos, el pipeline que més s'assembla al resultat obtingut pel pipeline original, és la versió que utilitza la base de dades de Kraken de 50GB. Tot i això, el pipeline que utilitza la base de dades de 5GB també aconsegueix uns resultats força interessants, considerant el temps d'execució que s'ha pogut estalviar a l'hora d'utilitzar aquest pipeline.

És cert que Bracken, és una eina que ha permès executar el pipeline en poc menys de 30 minuts (en el cas de les llibreries Mixed-Species). Si més no, amb aquest pipeline si que es perd una part més important d'informació (arribant fins a perdre el 10% de les mostres d'una espècie), i per tant, de moment és una eina que és recomana utilitzar únicament si es pot permetre la pèrdua d'informació per algun motiu en concret.

6 Conclusions

Finalment, ara que ja s'ha acabat aquest treball de final de grau, se'n poden extreure una sèrie de conclusions.

Per una banda, en quant als objectius proposats inicialment, es pot considerar que s'han assolit si no tots, la gran majoria d'aquests. Per una banda, ha estat un treball força educatiu, que ens ha permès crèixer com a professionals.

Per una altra banda, es poden considerar els objectius més específics del treball (intentar trobar alguna alternativa per intentar quantificar mostres d'ADN de la forma més eficient possible i sintetitzar la feina i els resultats obtinguts) com a força assolits ja que tot i no haver aconseguit una millora substancial en la reducció del temps d'execució, sí que s'ha pogut reduir lleugerament i entendre tant el per què, com el funcionament dels diferents algorismes i eines que han intervingut.

En quant als resultats obtinguts, s'ha pogut comprovar, que l'ús de Kraken pot ajudar a incrementar lleugerament l'eficiència del pipeline original. Malgrat aquesta petita millora, s'ha pogut observar que realment l'estratègia d'utilitzar kraken2 per reduir les mostres d'entrada, no és la millor opció, ja que arribarà un punt en que kraken classificarà una quantitat molt semblant de mostres a l'original.

Per aquest motiu, de cara a un futur, a mesura que vagin treballant en l'eina Bracken, seria interessant tornar a executar aquest pipeline i observar quina quantitat d'informació s'arriba a perdre amb les noves versions, ja que de no ser molta, es podria eliminar l'ús dels algorismes BWA i $\gamma - \delta$ i obtenir resultats en un temps molt inferior a l'actual.

6.1 Futures línies d'investigació

De cara a un futur, hi ha vàries coses que es podrien intentar millorar (ja sigui a nivell d'implementació d'un algorisme, com a nivell d'utilització d'altres màquines).

Un aspecte interessant que s'hauria d'intentar, és executar els pipelines en màquines més potents, amb més RAM disponible, i amb capacitat per utilitzar més threads de manera que l'efecte d'haver de carregar la taula hash que necessita l'alineador Kraken per funcionar, sigui menor i això es trobi reflectit en els temps d'execució finals.

Un altre aspecte que es podria intentar, és buscar la manera de dividir el pipeline en tantes parts com nodes es vulguin utilitzar per reduir el tamany del problema i que cada node pugui executar una part més petita del pipeline en qüestió (aplicant la tècnica de "Divide & Conquer").

De cara a treballs futurs relacionats amb aquesta temàtica, s'ha generat un repositori de Github[11] en el que s'hi troben els scripts més importants generats al llarg d'a-

quest treball, de manera que pugui servir de guia a altres persones en cas de necessitar-ho.

AGRAÏMENTS

Agraïments especials al tutor d'aquest treball Miquel Àngel Senar per la seva implicació en el seguiment del mateix, els consells donats en aquest període de temps i el bon ambient creat des del primer dia.

Per una altra banda agraïr també a la Lidia Garrido per permetre'm treballar a partir d'un dels seus estudis (incloent tant documentació com un pipeline implementat per ella).

I finalment agraïr a la meva família pel suport i els ànims constants i per haver-me permès realitzar aquests estudis.

REFERÈNCIES

[1] Lidia Garrido-Sanz, Miquel Àngel Senar, and Josep Pinol: Estimation of the relative abundance of species in artificial mixtures of insects using low-coverage shotgun metagenomics, 2020. [Online] Available: https://mbmg.pensoft.net/article/48281/

[2] Manual Reference Pages - Burrows-Wheeler Alignment Tool (March 8, 2013). [Online] Available: http://bio-bwa.sourceforge.net/bwa.shtml

[3] Manual page from samtools-1.15 - Samtools View (February 21, 2022). [Online] Available: http://www.htslib.org/doc/samtools-view.html

[4] Johns Hopkins University, Center for Computational Biology. Kraken2: Taxonomic Sequence Classification System (October 3, 2019). [Online] Available: https://ccb.jhu.edu/software/kraken2/

[5] Johns Hopkins University, Center for Computational Biology. Bracken: Bayesian Reestimation of Abundance with Kraken (August 7, 2019). [Online] Available: https://ccb.jhu.edu/software/bracken/

[6] Lidia's Gamma-Delta Algorithm repository (July 30, 2020). [Online] Available: https://github.com/LidiaGS/g-d_algorithm

[7] Derrick Wood, Kraken2 github repository (May 10, 2021). [Online] Available: https://github.com/DerrickWood/kraken2

[8] Jennifer Lu, KrakenTools github repository (April 30, 2020). [Online] Available: https://github.com/jenniferlu717/KrakenTools

[9] Jennifer Lu, Bracken github repository (March 21, 2021). [Online] Available: https://github.com/jenniferlu717/Bracken

[10] SchedMD. Slurm workload manager Version 21.08 documentation (February 24, 2022). [Online] Available: https://slurm.schedmd.com/

[11] Esteve Pineda, DNA_Quantification github repository (Important scripts generated during the last period). [Online] Available: https://github.com/EstevePineda/DNA_Quantification/

APÈNDIX

A.1 Taules completes

En aquesta secció de l'apènix s'adjunten algunes taules necessàries per poder entendre els resultats obtinguts en executar els diferents pipelines però que degut al seu tamany no s'han pogut afegir en el cos de l'informe.

TAULA 2: ABUNDÀNCIA D'ESPÈCIES OBTINGUDA EN EXECUTAR ELS DIFERENTS PIPELINES (MIXEDSPPLIB)

Abundància d'especies (Mixed Spp Lib)						
	Lib. 1	Lib. 2	Lib. 3	Lib. 4	Lib. 5	Lib. 6
Pipeline Original	AE (0.64971) BO (0.17167) BT (0.06582) AM (0.05033) DMo (0.02998) DMe (0.01052)	AE (0.37924) BO (0.37805) AM (0.12349) BT (0.03834) DMe (0.02339) DMo (0.01707)	AE (0.44486) BO (0.16181) AM (0.10563) BT (0.08595) DMo (0.0789) DMe (0.03823) LH (0.0354) AP (0.03213)	AE (0.30977) BO (0.2496) AM (0.15797) BT (0.06737) DMo (0.05985) DMe (0.05742) LH (0.0536) AP (0.01873)	AE (0.32342) BO (0.1536) AM (0.13569) DMo (0.12864) BT (0.09288) DMe (0.0694) AP (0.06842) PM (0.01576)	AE (0.27894) BO (0.18874) AM (0.16409) DMo (0.10473) DMe (0.08641) BT (0.082) AP (0.06055) PM (0.01866)
Bracken	AE (0,54644) BO (0,21129) BT (0.10248) AM (0,06602) DMo (0,02700) DMe (0,01373)	AE (0,28843) BO (0,41937) AM (0,14641) BT (0,05392) DMe (0,02742) DMo (0,01398)	AE (0,3523) BO (0,18697) AM (0,12956) BT (0.12689) DMo (0.06640) DMe (0.04617) LH (0.02886) AP (0.02770)	AE (0.23180) BO (0.27309) AM (0.18353) BT (0.09419) DMo (0.04778) DMe (0.06599) LH (0.04145 AP (0.01561)	AE (0.24216) BO (0.16789) AM (0.15700) DMo (0.10282) BT (0.13077) DMe (0.07935) AP (0.05614) PM (0.04038)	AE (0.20438) BO (0.20055) AM (0.18463) DMo (0.08104) DMe (0.09499) BT (0.11107) AP (0.04814) PM (0.04751
Kraken 5G + bwa + gd	AE (0.64711) BO (0.16217) BT (0.0644 1) AM (0.04396) DMo (0.02706) DMe (0.0104)	AE (0.37822) BO (0.37022) AM (0.11038) BT (0.03752) DMe (0) DMo (0)	AE (0.44547) BO (0.15459) AM (0.09435) BT (0.08563) DMo (0) DMe (0) LH (0.03486) AP (0.01682)	AE (0.31104) BO (0.24282) AM (0.14188) BT (0.067) DMo (0) DMe (0) LH (0.05361) AP (0.01682)	AE (0.32496) BO (0.14528) AM (0.12094) DMo (0.12153) BT (0.09265) DMe (0.07078) AP (0.06315) PM (0.01314)	AE (0.28071) BO (0.18518) AM (0.14831) DMo (0.09966) DMe (0.0889) BT (0.08277) AP (0.05723) PM (0.01653)
Kraken 50G + bwa + gd	AE (0.64804) BO (0.16023) BT (0.06311) AM (0.04756) DMo (0.0278) DMe (0.01016)	AE (0.37779) BO (0.35797) AM (0.11814) BT (0.03631) DMe (0) DMo (0)	AE (0.44405) BO (0.14964) AM (0.10086) BT (0.0825) DMo (0) DMe (0) LH (0.03401) AP (0.02894)	AE (0.30969) BO (0.23851) AM (0.15266) BT (0.06524) DMe (0) DMo (0) LH (0.05278) AP (0.01717)	AE (0.32309) BO (0.14235) AM (0.1302) DMo (0.1236) BT (0.08996) DMe (0.06879) AP (0.06337) PM (0.01277)	AE (0.2779) BO (0.1762) AM (0.15763) DMo (0.09957) DMe (0.08563) BT (0.07863) AP (0.05616) PM (0.01542)

TAULA 3: COMPARATIVA ENTRE PIPELINES EN FUNCIÓ DELS READS ASSIGNATS (SINGLE SPECIES)

Comparativa entre pipelines en funció dels reads assignats (Single Spp lib)			
	Reads inicials	Reads Classificats (kraken2)	Reads Assignats ($\gamma - \delta$)
Pipeline Original	24771346	—	11256431
Kraken2 5G + bwa + gd	24771346	20132549	10588989
Kraken2 50G + bwa + gd	24771346	23038511	11200616

TAULA 4: COMPARATIVA ENTRE PIPELINES EN FUNCIÓ DELS READS ASSIGNATS (MIXED SPECIES)

Comparativa entre pipelines en funció dels reads assignats (Mixed Spp lib)			
	Reads inicials	Reads Classificats (kraken2)	Reads Assignats ($\gamma - \delta$)
Pipeline Original	10100512	—	5344702
Kraken2 5G + bwa + gd	10100512	8493665	4956550
Kraken2 50G + bwa + gd	10100512	9685751	5305721

TAULA 5: TEMPS D'EXECUCIÓ DELS DIFERENTS PIPELINES I LLIBRERIES (EN HORES)

Temps execució pipeline complet [en hores]				
Llibreries	Pipeline original	Kraken (5GB) + BWA + gd	Kraken (50GB) + BWA + gd	Kraken (5GB) + Bracken
Single Species	46,93	43,02	44,05	0,754024
Mixed Species	20,41	17,42	20	0,371