BowBin: una pipeline per l'estrazione di virus da dati metagenomici correlati a IBS

Alessandro Aquino and *Nicolapio* Gagliarde

Abstract. IBS è il disturbo gastrointestinale funzionale più comune. La diagnosi di IBS si basa sui seguenti sintomi: cambiamenti nelle abitudini intestinali, dolore addominale e crampi. La causa di questo disturbo è sconosciuta e probabilmente dovuta a più fattori tra i quali i virus. Data quindi la frequenza di IBS, il tipo di sintomatologia e il ruolo fondamentale che il viroma potrebbe avere come possibile causa, abbiamo deciso di creare una pipeline, BowBin, per la ricerca di virus che potrebbero essere associati ad IBS. La pipeline prende in input dei dati metagenomici, prelevati da persone affette, e un insieme di genomi virali. Successivamente, grazie a Bowtie2, si esegue l'allineamento tra gli scaffolds estratti dai genomi virali e le reads dei dati metagenomici. Al seguito dell'allineamento è possibile utilizzare MetaBat2 e lo script python di vRhyme per generare la coverage table. Infine la tabella viene data in input a Binsanity con lo scopo di clusterizzare gli scaffolds ed analizzare la loro classificazione tassonomica. I risultati ottenuti mostrano che tutti i bin raggruppano i virus fino alla classe *Caudoviricetes*. Inoltre si può anche notare la presenza frequente di alcuni virus come: *Achromobacter phage B_AchrS_AchV4*, vari *Lactococcus phage* e *Xanthomonas phage f20Xaj*. Infine risulta evidente che la totalità dei virus rilevati è batteriofaga. Inoltre la nostra pipeline, BowBin, è stata confrontata con Genome Detective, i risultati mostrano virus rilevati da entrambe le pipeline, come: *Lactococcus phage* e *Skunavirus*, alcuni virus sono stati rilevati con varianti diverse ma molto vicine geneticamente, e altri virus sono stati rilevati da Genome Detective ma non da BowBin, questo potrebbe essere dovuto al numero di genomi virali limitato dati in input alla nostra pipeline.

Introduzione

0.1 Problema

Irritable Bowel Syndrome (IBS) è il disturbo gastrointestinale funzionale più comune e colpisce circa l'11% della popolazione mondiale[1]. La diagnosi di IBS si basa sui seguenti sintomi: cambiamenti nelle abitudini intestinali, dolore addominale e crampi[2]. La causa di questo disturbo è sconosciuta e probabilmente dovuta a più fattori come ansia, depressione [3], dieta [4], infiammazioni[5] e alterazioni del batterioma intestinale[6].
Siccome il viroma intestinale è composto principalmente da virus batteriofagi, la composizione del batterioma dipende fortemente dalla co-evoluzione, dal trasferimento orizzontale dei geni, ma soprattutto dalla predazione dei batteri da parte dei virus[7].

0.2 Pipeline: BowBin

Data quindi la frequenza di IBS, il tipo di sintomatologia e il ruolo fondamentale che il viroma potrebbe avere come possibile causa, abbiamo deciso di creare una pipeline, BowBin, per la ricerca di virus che potrebbero essere associati ad IBS. La pipeline prende in input dei dati metagenomici, prelevati da persone affette, e un insieme di genomi virali. Successivamente, grazie a Bowtie2[8], si esegue l'allineamento tra gli scaffolds estratti dai genomi virali e le reads dei dati metagenomici. Al seguito dell'allineamento è possibile utilizzare MetaBat2[9] e lo script python di vRhyme[10] per generare la coverage table. Dall'analisi della coverage table è possibile capire i virus, con il relativo numero di scaffolds, presenti nei dati metagenomici. Infine la tabella viene data in input ai due tool di binning: vRhyme e Binsanity[11] con lo scopo di clusterizzare gli scaffolds ed analizzare la loro classificazione tassonomica.

0.3 Risultati

I risultati ottenuti che verranno descritti nel dettaglio nella sezione 4 del paper, con relativi grafici e tabelle, mostrano che tutti i bin raggruppano i virus fino alla classe *Caudoviricetes*. Inoltre si può anche notare la presenza frequente di alcuni virus come: *Achromobacter phage B_AchrS_AchV4*, vari *Lactococcus phage* e *Xanthomonas phage f20-Xaj*. Infine risulta evidente che la totalità dei virus rilevati è batteriofaga.

BowBin è stata confrontata con Genome Detective[12], i risultati mostrano virus rilevati da entrambe le pipeline, come vari *Lactococcus phage* e *Skunavirus*, alcuni virus sono stati rilevati con varianti diverse ma molto vicine geneticamente, e altri virus sono stati rilevati da Genome Detective ma non da BowBin, questo potrebbe essere dovuto al numero di genomi virali limitato dati in input alla pipeline qui descritta.

0.4 Contributi di BowBin

Il contributo utile che fornisce BowBin è la replicabilità; necessitando di poche risorse hardware e di memoria è possibile eseguirlo anche su componenti non particolarmente prestanti. Soprattutto grazie all'utilizzo di Bowtie2 e alla creazione dell'indice, che può essere generato una sola volta e utilizzato infinite volte per l'allineamento con le reads metagenomiche. BowBin, inoltre, estende il binning e la rilevazione non solo di virus ma anche batteri e funghi, rendendosi anche versatile non limitandosi ai virus.

1 Related work

In questa sezione verrano presentanti papers che hanno delle similitudini con il nostro elaborato; sia per quanto riguarda l'obiettivo e sia per alcuni tool usati nel processo produttivo.

1.1 VirusSeeker, a computational pipeline for virus discovery and virome composition analysis

Ci sono due categorie di pipeline che permettono l'elaborazione di sequenze microbiche: quelle progettate per analizzare la composizione del viroma descrivendo l'abbondanza e le specie di virus presenti, e quelle progettate per scoprire nuovi virus. Nel paper [13] viene presentata la pipeline VirusSeeker-Virome (VS-Virome). VS-Virome è in grado di definire il tipo e l'abbondanza delle sequenze virali, sia quelle già conosciute alla comunità scientifica che non. Inoltre, per la scoperta di nuovi virus, viene presentata una variante di VS-Virome, chiamata VS-Discovery. Di seguito viene descritta solo la pipeline VS-Virome, in quanto è più vicina al problema affrontato in questo lavoro.

1.1.1 VS-Virome workflow

VS-Virome riceve in input un file FASTQ, quindi, la fase di pre-processing inizia con il fondere una coppia di reads in una singola read usando fastq-join[14], poi applica un filtro di qualità alle reads ottenute, usando il tool PRINSEQ [15]. Per ridurre il carico computazionale è stata eseguita la clusterizzazione delle reads molto simili ($\geq$ 95%) con CD-HIT [16]. Viene applicato un altro filtro qualità che scarta le reads che non contengono tratti di almeno 50 nucleotidi diversi da "N" consecutivi. Per concludere questo step vengono usati i due tool: MegaBLAST e BWA-MEM [17] per escludere le sequenze genomiche dell'ospite (in questo caso un esemplare adulto di *Macaca mulatta*), che quindi vengono classificate come "non virali". Concluso lo step di pre-processing si passa alla fase di sequenziamento. Le reads ottenute dallo step precedente vengono allineate utilizzando BLASTn con un database che contiene solo sequenze nucleotidiche virali, in questo modo è possibile identificare le reads che hanno seqeunze nucleotidiche in comune con virus già conosciuti. Le restanti reads vengono allineate usando BLASTx con un database che contiene le sequenze proteiche dei virus, in questo modo è possibile identificare le reads che hanno sequenze proteiche in comune con virus già conosciuti. In base al punteggio ottenuto dai due step appena descritti, una reads può essere classificata come "non virale", "phage" (quindi indica un virus batteriofago), oppure può essere classificata come "candidate viral sequences". Nell'insieme denominato "candidate viral sequences" vengono inserite le reads che potrebbero appartenere a virus eucarioti, quindi vengono allineate usando MegaBLAST e il database NCBI NT, in base al punteggio ottenuto, una reads viene data o in pasto alla classificazione finale o viene allineata usando BLASTn con il database NCBI NT. Di nuovo, in base al punteggio ottenuto, una reads viene data o in pasto alla classificazione finale o viene allineata usando BLASTx con il database NCBI NR, in base al punteggio ottenuto da quest'ultimo step, la reads o viene data in pasto alla classificazione finale o viene classificata come "Unassigned".

Il processo di classificazione finale si basa sul database tassonomico di NCBI, questo processo riceve una reads e la classifica in "Eukaryotic virus", "Phage", "Ambiguous" oppure "Non-viral".

1.1.2 Risultati

La pipeline VS-Virome permette di analizzare più campioni usando un solo comando, inoltre gode di un alto throughput ed è completamente automatizzata. Grazie alla sua modularità può essere modificata in base al problema.

VS-Virome è stata testata su un campione di dati estratto da scimmie affette da SIV (Virus di Immunodeficienza delle scimmie). Dal campione sono state ottenute 154,522 reads uniche, 8053 sono state classificare come sequenze batteriofaghe mentre 2155 sono state classificate come "candidate viral sequences". Di queste 2155, la pipeline ha rilevato 574 sequenze virali e 66 ambigue. Le restanti reads sono state assegnate a regni diversi da quello dei virus oppure a virus batteriofagi. Purtroppo la maggrior parte delle reads classificate come sequenze virali, nello specifico il 59.3% delle 574 sequenze, si è rilevato più simile a sequenze appartenenti a funghi e batteri. Quindi VS-Virome ha presentanto un tasso di falsi positivi del 59.3%.

VS-Virome è stata confrontata con VirFind [18]. VirFind da rilevato 66 sequenze virali con BLASTn e 1642 con BLASTx. Dall'output di VirFind, sono state poi selezionate le prime 20 reads (in ordine di punteggio) del report BLASTx, e utilizzando il motore di ricerca dei database NCBI NR, gli autori di VS-Virome, hanno riscontrato che la totalità delle reads è più vicina a sequenze di batteri, e non a virus eucarioti. Quindi i falsi positivi di VirFind, su questo piccolo insieme, ammontano al 100%. La procedura appena descritta è stata effettuata anche per i dati ottenuti da VS-Virome ed è stato notato un significante calo del falsi positivi.

1.2 Classification and quantification of bacteriophage taxa in human gut metagenomes

Nel paper [19] si studia la divesità dei virus batteriofagi nell'intestino umano, partendo da campioni metagenomici ma non solo. Vengono estratti anche i metagenomi dall'oceano per poi confrontarne le famiglie e trovare delle somiglianze. In questa sezione verrà descritta la pipeline utilizzata per la classificazione dei virus nell'intestino umano, dato che è una problematica simile alla nostra. In questo studio vengono sviluppati due webserver per la classificazione dei virus: VIROME che assegna le reads virali a livello di regno (virus, batteri ...) e MetaVir invece assegna i geni alle famiglie virali sulla base di un elenco di 12 geni marcatori per ampie famiglie virali. Vengono usati questi geni marcatori per classificare tassonomicamente e quantificare i taxa fagici contenuti all'interno di 252 metagenomi pubblicati derivati da campioni fecali di 207 individui. La classificazione tassonomica derivata dei profagi è stata impiegata per dedurre una vasta rete di interazioni profago-ospite all'interno del microbioma intestinale.

1.2.1 Analisi dei metagenomi intestinali

Complessivamente, 252 campioni metagenomici, da 207 individui, ottenuti dal progetto MetaHIT[20] (71 danesi, 39 spagnoli; tutti campionati una volta), il progetto NIH Human Microbiome[21] (94 individui statunitensi; 51 individui campionati una volta, 41 campionati due volte e 2 campionati tre volte) e la Washington University (tre campioni statunitensi; tutti campionati una volta) sono stati analizzati. La raccolta dei campioni e l'estrazione del DNA per i campioni MetaHIT e Human Microbiome Project hanno seguito i rispettivi protocolli. Le reads della sequenza Illumina sono state elaborate utilizzando MOCAT[22] dove le reads sono state assemblate in scaffolds e sono stati rilevati i geni. I geni sono stati quindi raggruppati utilizzando CD-HIT-EST[16] per creare un catalogo di geni di riferimento, qui chiamato catalogo 252refGene. Quindi, l'abbondanza di ciascuno di questi geni in ciascun campione è stata determinata mappando le reads metagenomiche da ciascun campione a ciascun gene di riferimento utilizzando SoapAligner 2.21 (lunghezza minima di reads 45 nt), e quindi dividendo la copertura per coppia di basi per la lunghezza del gene (bp). Per determinare la tassonomia dell'ospite batterico per i profagi scaftig, sono state impiegate due tecniche separate utilizzando le sequenze scaftig, vale a dire un metodo di classificazione dei nucleotidi e BlastN contro genomi batterici di riferimento. L'identità dell'ospite è stata assegnata al taxon batterico più specifico.

1.2.2 Risultati

Questa analisi ha portato all'identificazione di otto generi virali all'interno dei campioni di viroma, sette dei quali sono stati identificati anche nel catalogo del metagenoma totale di 252, ad eccezione dei virus simili a *FelixO1*. Inoltre, il fatto che i *Gokushovirinae*, che appartengono alla famiglia dei piccoli virus *Microviridae*, siano abbondanti nei viromi così come nel catalogo di 252 metagenomi totali indica che l'estrazione del campione di DNA metagenomico totale non esclude questi piccoli virus contrariamente a suggerimenti precedenti.

1.3 Development of a virus detection and discovery pipeline using next generation sequencing

La pipeline di questo paper [18] è capace di elaborare più campioni, rilevando virus noti e scoprendone di nuovi. Nello specifico il programma utilizza i dati NGS per identificare virus noti e sconosciuti e fornire una solida pipeline per l'utente finale.

1.3.1 Workflow

Il primo step è quello di prelevare i campioni, in questo caso lo si è fatto da piante (31 campioni) infette. Successivamente da questi campioni viene effettuato il sequenziamento. Si passa allo sviluppo di VirFind utilizzato per eleborare gli outuput di NGS. Il workflow inizia prendendo i file di sequenza NGS che vengono convertiti in file fasta. Le sequenze vengono quindi ritagliate alle estremità 5' e 3' per rimuovere eventuali adattatori e primer. Le sequenze vengono poi mappate ai genomi di riferimento usando Bowtie2. Poi l'assemblaggio viene rieffettuato per reads non mappate usando Velvet[23]. Originariamente VirFind è stato costruito per trovare i virus delle piante. Tuttavia, poiché la scoperta dei virus mediante il confronto delle sequenze è la stessa indipendentemente dalla specie o dall'ospite del virus e con il fatto che VirFind è stato testato con successo anche con i virus delle api, lo strumento può essere utilizzato per il rilevamento e la scoperta di virus in qualsiasi host.

1.3.2 Risultati

VirFind ha rilevato molti virus noti ma anche la scoperta di uno nuovo ossia, un nuovo *Trichovirus* è stato scoperto nel *Ribes nigrum* (ribes nero). Inoltre VirFind è stato testato anche con altri dati siccome è stato reso pubblico (https://virfind.org/j/) e quindi utilizzato da altri membri della comunità scentifica producendo risultati di rilevamento/scoperta di virus identici o migliori rispetto a quelli precedentemente identificati, a dimostrazione della riproducibilità dello strumento. Nel loro insieme, i risultati hanno dimostrato che con VirFind, il rilevamento e la scoperta dei virus mediante NGS possono essere standardizzati e facilmente accessibili a un pubblico più ampio di scienziati in assenza di un bioinformatico designato.

1.4 Genome Detective: an automated system for virus identification from high-throughput sequencing data

Nel paper [12] si descrive la pipeline Genome Detective, progettata per identificare e classificare in modo rapido e

accurato i virus presenti in dati NGS. La pipeline riceve in input dei file FATSQ su cui applica dei filtri per le reads di bassa qualità ed esegue il processo di trimming con Trimmomatic [24]. Prima e dopo del trimming viene visualizzata la qualità delle reads usando FastQC [25]. Per rilevare le reads virali viene usato il tool di allineamento DIAMOND[26], che si basa sulle proteine. Per migliorare la velocità e la sensibilità, gli autori di Genome Detective hanno usato il database di proteine Swissprot UniRef90[27], che contiene 494 134 cluster di proteine. Ogni cluster di proteine ha un ID associato al database NCBI RefSeq[28]. La pipeline ha la capacità di scaricare automaticamente le nuove versioni dei database.

Per incrementare la velocità e l'accuracy, grazie ai risulstati di DIAMOND, le reads vengono raggruppate in buckets. Ogni bucket ha lo scopo di raggruppare reads appartenenti allo stesso genoma virale. Ogni bucket viene quindi processato da SPAdes[29]. Successivamente viene utilizzato Blastx e Blastn per cercare delle sequenze di riferimento all'interno del database NCBI RefSeq. I contigs rilevati, di una specifica specie di virus, vengono uniti con Advanced Genome Aligner (AGA)[30].

Genome Detective è stato testato su 208 dataset, dimostrando una concordanza di oltre il 95% con quanto riportato nel Sequence Read Archive, identificando con successo 257 specie di virus. La pipeline è stata confrontata con IVA[31] e drVM[32]: rispetto ad IVA, Genome Detective si è mostrata più veloce di un fattore 10 nell'assemblare il genoma del virus HIV-1 e ha fornito contigs più lunghi e più accurati; rispetto a drVM, Genome Detective si è mostrata più accurata nella creazione dei contigs.

1.5 Analogie e differenze con BowBin

1.5.1 VirusSeeker, a computational pipeline for virus discovery and virome composition analysis

VS-Virome, a differenza di BowBin, applica una fase di pre-processing molto più elaborata, infatti raggruppa le reads simili ed esclude quelle associate all'host. Inoltre per classificare ogni reads utilizza cinque database: Virus-only nt DB, Virus-only protein DB, NCBI taxonomy DB, NCBI NR DB e NCBI NT DB, quest'ultimo viene utilizzato sia con BLASTn che con MegaBLAST. Data la presenza di tutti questi passi, VS-Virome potrebbe presentare un efficacia maggiore, ma con una velocità di esecuzione bassa. BowBin invece, siccome non utilizza interi database, ma si basa su dati scelti dall'utente, è più scalabile. Dato che possiamo usare più o meno dati in base al problema ma anche concentrarci su dati che non sono prettamente di virus. Tuttavia BowBin non prevede l'utilizzo di sequenze proteiche.

1.5.2 Classification and quantification of bacteriophage taxa in human gut metagenomes

In "Classification and quantification of bacteriophage taxa in human gut metagenomes" viene trattata la nostra stessa problematica si studia la divesità dei virus batteriofagi nell'intestino umano, partendo da campioni metagenomici. Si differenzia da BowBin per il raggruppamento dei geni e per l'utilizzo degli scaftig.

1.5.3 Development of a virus detection and discovery pipeline using next generation sequencing

A livello di tools utilizzati troviamo una similutidine ossia Bowtie2. In entrambe le pipeline, il tool viene utilizzato per allineare le sequenze metagenomiche con i genomi di riferimento. Nel caso di VirFind le sequenze metagenomiche sono state estratte da 31 piante, nel caso di BowBin inveve sono state estratte dall'intestino umano.

Inoltre sia VirFind che BowBin sono utilizzabili per rilevare organismi di regni diversi, infatti le due pipeline sono utilizzabili per il rilevamento di virus, batteri o funghi in qualsiasi host.

1.5.4 Genome Detective: an automated system for virus identification from high-throughput sequencing data

Le principali differenze tra Genome Detective e BowBin si possono riassumere in tre punti: il processo di trimming viene effettuato usando Trimmomatic da Genome Detective, invece BowBin non prevede questo step; Genome Detective utilizza DIAMOND per l'allinemaneto delle reads, invece BowBin utilizza Bowtie2; infine BowBin si basa su sequenze nucletotidiche fornite dall'utente, invece l'altra pipeline utilizza un database di proteine.

Entrambe le pipeline presentano un alta velocità di esecuzione: BowBin grazie a Bowtie2 e alla creazione dell'indice, Genome Detective grazie al raggruppamento delle reads in buckets.

2 Background

Per capire a fondo le tecniche e i problemi descritti nei capitoli successivi è indispensabile fornire al lettore alcune definizioni.

2.1 Irritable Bowel Syndrome(IBS).

La sindrome dell'intestino irritabile (IBS) è un disturbo gastrointestinale funzionale prolungato e invalidante con un tasso di incidenza del 11% nel mondo. L'IBS potrebbe compromettere seriamente la vita dei pazienti e causare un onere economico elevato per la comunità. La fisiopatologia dell'IBS è poco conosciuta, mentre diversi possibili meccanismi, come l'ipersensibilità viscerale, la motilità intestinale irregolare, le relazioni cervello-intestino anormali e il ruolo degli agenti infettivi, sono implicati nell'inizio e nello sviluppo di questa sindrome[6]. Diversi studi hanno dimostrato un'alterazione delle concentrazioni di linfociti B, mastociti (MC), linfociti T e citochine nella mucosa intestinale o nella circolazione sistemica che possono contribuire alla formazione dell'IBS. Pertanto, l'IBS potrebbe essere sviluppato in quelli con predisposizione genetica. Il ruolo delle infezioni nell'insorgenza e

nell'esacerbazione dell'IBS è stato indagato da numerosi studi clinici; è stato inoltre descritto il possibile ruolo di alcuni patogeni nello sviluppo e nell'esacerbazione di questa malattia. Sembra che i principali patogeni obbligati corrispondano alla malattia IBS, *Clostridium difficile*, *Escherichia coli*, *Mycobacterium avium*, *Campylobacter concisus*, *Campylobacter jejuni*, *Chlamydia trachomatis*, *Helicobacter pylori*, *Pseudomonas aeruginosa*, *Salmonella spp*, *Shigella spp*. Sulla base delle attuali conoscenze, lo studio attuale conclude che i più comuni agenti patogeni batterici, virali e parassitari possono essere coinvolti nello sviluppo e nella progressione dell'IBS. [33]

2.2 Materiali e tools

2.2.1 Coverage table

Il termine "coverage" descrive la relazione tra le reads e un genoma di riferimento che può essere completo o parziale. Nel nostro caso le reads sono contenute in file di dati metagenomici prelevati dall'intestino umano. E il genoma completo dei virus viene diviso in scaffolds.

La coverage, indicata con C viene calcolata seconda la seguente formula:

$$C = \frac{L * N}{G}$$

Dove L indica la lunghezza delle reads, N il numero delle reads allineate allo scaffold e G la lunghezza dello scaffold. La figura 1 riporta uno schema grafico che riassume concettualmente la formula.

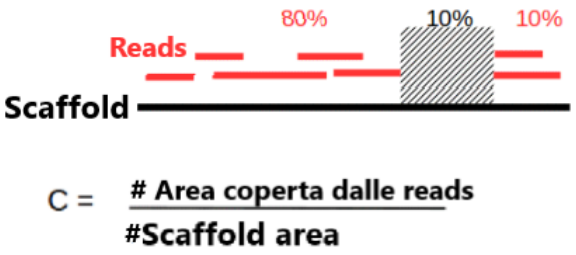

Figure 1. Calcolo della coverage

In questo lavoro la coverage table verrà generata utilizzando Metabat e verrà chiamata "depth.txt". La tabella conterrà la prima colonna che riporta il nome dello scaffold, nella seconda colonna sarà presente la sua lunghezza, la terza colonna è data dalla somma delle medie riportate nelle colonne successive, le ultime due colonne fanno riferimento al file metagenomico, la prima colonna contiene la coverage e la seconda la deviazione standard.

Il file "depth.txt" verrà poi processato da uno script che eliminerà la seconda e la terza colonna, moltiplicherà la colonna della coverage per una certa somma data in input alla pipeline, e applicherà la seguente formula alla colonna della deviazione standard:

$$Stdev_1 = (Stdev^{0,5}) * x$$

Dove $Stdev$ indica la deviazione standard presente nel file depth.txt, x è un valore dato in input alla pipeline e $Stdev_1$ è la deviazione standard che verrà salvata nella nuova coverage table. I passi appena descritti, con le relative variabili e operazioni, sono necessari per settare il tool di binning.

2.2.2 ViruSpy

ViruSpy è una pipeline progettata per la scoperta di virus dai dati di sequenziamento metagenomico disponibili nel database SRA dell'NCBI. Il primo passaggio identifica le reads virali nel campione metagenomico con Magic-BLAST[34], che consente questo passaggio senza dover scaricare il set di dati metagenomici (spesso piuttosto grandi). Le reads grezze estratte vengono assemblate in contigs e annotate per i geni da Glimmer[35]. Dopo l'annotazione, l'algoritmo Building Up Domains (BUD) ci consente di stabilire se i genomi virali sono non nativi (cioè integrati) in un genoma ospite.

Workflow

La pipeline di ViruSpy richiede all'utente di fornire l'ID SRA del campione metagenomico da ricercare e un database del genoma virale di riferimento. Il database del genoma virale di riferimento può essere fornito dall'utente sotto forma di un file FASTA o di un database BLAST. Se nessuno dei due viene fornito, ViruSpy passerà automaticamente al database del genoma virale RefSeq e tenterà di scaricare quelle sequenze in formato FASTA. Nella prima fase Magic-BLAST restituisce tutte le sequenze simili a virus dal campione SRA, che vengono assemblate in contig utilizzando l'assemblatore MEGAHIT[36]. I contigs vengono verificati come sequenze virali attraverso una previsione di frame di reads aperti all'interno dei contigs utilizzando Glimmer3[35]. I domini conservati virali (CD) sono determinati utilizzando il database NCBI CDD. I file di output vengono quindi combinati per identificare una serie di contig virali ad alta affidabilità. La determinazione delle reads endogene all'interno di un host si basa sull'algoritmo Building Up Domains (BUD). BUD prende come input un contig virale peprocessato identificato da un dataset di metagenomica e invia le estremità del contig da entrambi i lati a Magic-BLAST, che cerca reads sovrapposte nel dataset SRA. Le reads vengono quindi utilizzate per aggiungere il contig in entrambe le direzioni. Questo processo continua fino a quando i domini non virali vengono identificati su entrambi i lati del contig virale originale, il che implica che il contig originale era endogeno

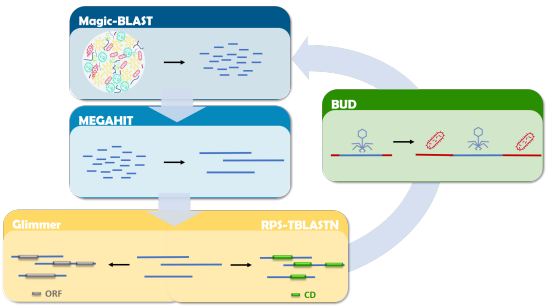

Figure 2. workflow ViruSpy

nell'host o fino a quando non viene raggiunto un numero specificato di iterazioni (il valore di iterazione predefinito era impostato su 10).Questo processo è illustrato di seguito:

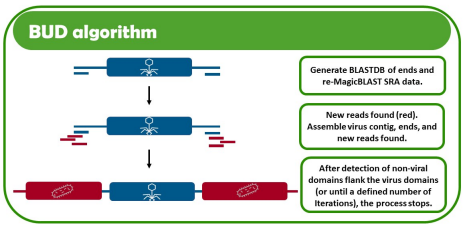

Figure 3. BUD

2.2.3 vRhyme

vRhyme è uno strumento multifunzionale per il binning dei genomi dei virus dai metagenomi. vRhyme funziona utilizzando i confronti della varianza di copertura e la classificazione supervisionata del machine learning delle caratteristiche della sequenza per costruire genomi virali metagenomici assemblati (vMAG). vRhyme è progettato per funzionare su sequenze/scaffold virali. Un flusso di lavoro tipico consiste nel prevedere i virus da un metagenoma e quindi utilizzare tali previsioni come input per vRhyme. vRhyme può prendere un intero metagenoma come input, ma le prestazioni per un intero metagenoma non sono state completamente valutate.vRhyme divente cruciale per tenere in considerazione la varianza di copertura che aiuta a separare gli scaffold che sembrano simili ma in realtà sono genomi diversi.Inoltre, nel suo benchmark si dimostra la sua velocità e accuratezza nel binning di scaffold virali, con basse richieste computazionali, nei metagenomi sintetici e naturali rispetto ad altri software di binning.

2.2.4 SAMtools

SAMtools è un insieme di utilità per l'interazione e la post-elaborazione di allineamenti di reads di brevi sequenze di DNA nei formati SAM (Sequence Alignment/Map) e BAM (Binary Alignment/Map). Questi file vengono generati come output da allineatori a reads breve come BWA . Vengono forniti strumenti sia semplici che avanzati, che supportano attività complesse come l'identificazione delle varianti e la visualizzazione dell'allineamento, nonché l'ordinamento, l'indicizzazione, l'estrazione dei dati e la conversione del formato. I file SAM possono essere anche molto grandi quindi la compressione viene utilizzata per risparmiare spazio. SAMtools consente di lavorare direttamente con un file BAM compresso, senza dover decomprimere l'intero file. Inoltre, poiché il formato di un file SAM/BAM è alquanto complesso (contiene reads, riferimenti, allineamenti, informazioni sulla qualità e annotazioni specificate dall'utente), SAMtools riduce lo sforzo necessario per utilizzare i file SAM/BAM nascondendo i dettagli di basso livello.

2.2.5 Bowtie2

Bowtie2 è uno tool efficiente in termini di memoria per allineare le reads di sequenziamento a lunghe sequenze di riferimento. È particolarmente efficace nell'allineare reads di circa 50 fino a 100 o 1.000 di caratteri, ed è particolarmente efficace nell'allineare genomi relativamente lunghi (ad es. mammiferi).

2.2.6 BinSanity

BinSanity è un tool utilizzato per il binning di genomi, basato sull'algoritmo di clustering Affinity Propagation (AP) e sulle informazioni di coverage. A differenza di altri algoritmi di clustering che possono raggruppare efficacemente frammenti di DNA correlati utilizzando dati di coverage, come il clustering gerarchico e k-means, BinSanity non richiede l'input umano di criteri informativi che determinano il numero finale di cluster (ad esempio, criterio informativo bayesiano). Infatti AP non richiede input per determinare i centri dei cluster; invece ogni punto è considerato iterativamente come un potenziale centro di cluster. Quanto appena detto rappresenta un notevole punto di forza, siccome assegnare un numero a priori per la diversità della comunità è sempre più difficile negli ecosistemi complessi.

2.2.7 Metabat

Metabat è pensato per il raggruppamento di grandi frammenti genomici assemblati da sequenze metagenomiche consentendo lo studio dei singoli organismi e delle loro interazioni. A causa della natura complessa di queste comunità, i metodi di binning del metagenoma esistenti spesso mancano di un gran numero di specie microbiche. MetaBAT integra le distanze probabilistiche empiriche dell'abbondanza del genoma e la frequenza del tetranucleotide per un accurato binning del metagenoma. Inoltre supera i metodi alternativi in termini di accuratezza ed efficienza computazionale su dataset metagenomici sia sintetici che reali. Forma automaticamente centinaia di contigs del genoma di alta qualità su un assemblaggio molto grande costituito da milioni di contig nel giro di poche ore su un singolo nodo.

2.3 Formato dei file utlizzati

- Il formato FASTA è un formato basato su testo per rappresentare sequenze nucleotidiche o sequenze di amminoacidi (proteine), in cui i nucleotidi o gli amminoacidi sono rappresentati utilizzando codici a lettera singola. Il formato consente inoltre ai nomi delle sequenze e ai commenti di precedere le sequenze.La semplicità del formato FASTA semplifica la manipolazione e l'analisi delle sequenze utilizzando strumenti di elaborazione del testo e linguaggi di scripting come Python.

- Il formato SAM consiste in un'intestazione e una sezione di allineamento. L'equivalente binario di un file SAM è un file Binary Alignment Map (BAM), che memorizza gli stessi dati in una rappresentazione binaria compressa. I file SAM possono essere analizzati e modificati con il software SAMtools, che può anche ordinarli con il comando "sam tool sort". Il tratto di testata deve essere precedente al tratto di allineamento se presente. Le intestazioni iniziano con il simbolo '@', che le distingue dalla sezione di allineamento. Le sezioni di allineamento hanno 11 campi obbligatori, oltre a un numero variabile di campi facoltativi.
- Il formato tsv è usato per la memorizzazione di dati in una struttura tabulare, ad esempio una tabella di database o dati di un foglio di calcolo. Ogni record nella tabella è una riga del file di testo . Ogni valore di campo di un record è separato dal successivo da un carattere di tabulazione . Il formato TSV è quindi una variazione del formato dei valori separati da virgole .

3 Implementazione BowBin

La nostra pipeline si basa sul ricercare virus che possono essere associati ad IBS partendo da metagenomi intestinali di persone affette. Inizialmente proviamo i tools ViruSpy e Vibrant per estrarre le reads dei genomi virali dai metagenomi. Questi due tools non hanno prodotto risultati attentibili, per tanto abbiamo deciso di scaricare direttamente i genomi virali da NCBI. Da questi genomi virali, utilizzando lo script "split.py", vengono creati gli scaffolds. Gli scaffolds verranno dati in input Bowtie2 per generare l'indice. Successivamente viene effettuato l'allineamento delle reads metegenomiche con l'indice, e quindi con gli scaffolds. Al seguito dell'allineamento è possibile utilizzare MetaBat2 e lo script python di vRhyme per generare la coverage table. Successivamente passiamo quest'ultima in input a vRhyme e come output avremmo dovuto ottenere il binning, così non è stato. Il tool BinSanity, sarà utilizzato al posto di vRhyme per effettuare il binning. Ottenendo così il nostro obiettivo cioè, dati dei genomi virali, capire la loro presenza nei dati metagenomici. Di seguito verrà mostrato il grafico della pipeline e i passaggi nel dettaglio.

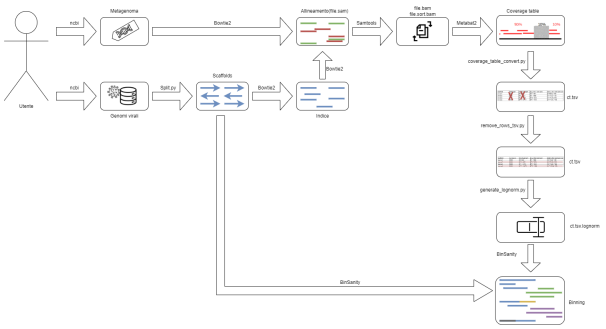

Figure 4. Rappresentazione della pipeline

3.1 Raccolta dei genomi virali

Per la raccolta dei genomi di virus si è deciso di testare i due tool: ViruSpy[37] e Vibrant[38]. Questi tool prendono in input dei dati metagenomici da cui estraggono i genomi virali. Purtroppo nessuno dei due tool ha prodotto l'output desiderato, infatti Vibrant ha bisogno di reads metagenomiche di almeno 1000 basi, e i dati presi in esame contengono reads di 250 basi in media. ViruSpy invece estrae meno di dieci contigs.

Vibrant è stato testato usando il seguente comando:

```
python3 ../VIBRANT_run.py -i ERR5084065.fasta
```

Abbiamo provato anche a mettere insieme più reads metagenomiche fino a superare la lunghezza di 1000bp ma Vibrant non ha prodotto comunque nessun output.

Per testare ViruSpy invece abbiamo prima creato un database BLAST:

```
makeblastdb -in viral.1.1.genomic.fna -dbtype nucl -parse_seqids -out databaseBlasta -title "nomedb"
```

Il comando makeblastdb con il flag -in produce un database BLAST in base al file viral.1.1.genomic.fna; il parametro -dbtype è usato per specificare il tipo di dato contenuto nel database, nel nostro caso acidi nucleici; mentre il flag -parse_seqids è necessario per mantenere gli identificatori di sequenza originali. Altrimenti makeblastdb genererà i propri identificatori. Successivamente si esegue ViruSpy sui dati metagenomici per estrarre le reads dei virus.

```
viruspy.sh -b databaseBlasta -s ERR80540XX -o outputFolder
```

Nello specifico il flag -b va a specificare che in input un database BLAST da utilizzare con Magic-Blast; -s per specificare il numero di accesso al database SRA (cioè l'ID del file che contiene i dati metagenomici); -o utilizzato per la cartella di output.

Siccome i due tool appena descritti non hanno prodotto un output utilizzabile dalla pipeline, abbiamo deciso di scaricare i genomi completi dei virus manualmente. Dato che il viroma dei pazienti affetti da IBS è composto principalmente da virus batteriofagi appartenenti all'ordine *Caudovirales* [39] abbiamo preso in considerazione 4494 virus appartenenti al regno *Duplodnaviria*.

3.2 Rilevazione dei virus

La prima operazione da effettuare è la divisione dei genomi virali in scaffolds, per automatizzare questa operazione utilizziamo l'apposito script:

```
python3 scripts/split.py genomiCompleti.fasta scaffoldsVirus.fasta 218
```

Il primo parametro è il file che contiene i genomi completi dei virus, il seconda parametro è il file in cui salvare gli scaffolds, il terzo parametro indica il numero di righe che compongono uno scaffold. In questo lavoro si è deciso di creare scaffolds composti da 218 righe, per un totale di circa 13mila basi nucleotidiche.

Per eseguire l'allineamento utilizzando Bowtie2 bisogna innanzitutto creare un indice. L'indice di riferimento sarà formato dagli scaffolds estratti dai genomi dei virus dati in input alla pipeline.

```
bowtie2-build scaffoldsVirus.fasta nomeIndice
```

Per evitare problemi dovuti alla versione di Bowtie2, si può sostituire il simbolo "@" con il simbolo ">" nei metagenomi con l'apposito script:

```
    python3 scripts/remove_at.py ERR50840XX.
    fasta ERR50840XX.fasta
```

Dopo di che vengono allineati gli scaffolds dell'indice con le reads contenute nei dati metagenomici.

```
bowtie2 --no-unal --no-discordant -f -x
    nomeIndice -U ERR50840XX.fasta -S nomesam.
    sam
```

Il flag –no-unal disabilita il salvataggio di reads che non sono state allineate, invece il flag –no-discordant disabilita la ricerca di allineamenti discordanti (questi due flag sono stati settati seguendo il paper di vRhyme[10]); Il flag -f permette di specificare un file .fasta dopo il flag -U; Il flag -U identifica il file che contiene le reads da allineare con l'indice specificato dopo -x; Il flag -S identifica il file .sam in cui salvare gli allineamenti.

Il file sam viene convertito in file bam

```
samtools view -S -b sample.sam > sample.bam
```

Il comando samtools view permette la stampa di un file sam; il flag -S permette di specificare il file .sam; il flag -b specifica il formato .bam in output.

Il file bam viene poi ordinato

```
samtools sort file.bam > fileSort.bam
```

Viene utilizzato "jgi_summarize_bam_contig_depths" di Metabat2 che, dato in input il file BAM ordinato, genera la coverage table e la salva nel file depth.txt

```
jgi_summarize_bam_contig_depths --outputDepth
    depth.txt fileSort.bam
```

A questo punto la coverage table viene modificata tramite lo script coverage_table_convert.py di vRhyme, che prende in input il file depth.txt e produce un file .tsv che contiene la coverage table.

```
coverage_table_convert.py -i depth.txt -o
    coverage_table.tsv -multiplyAvg 900 -
    multiplyStdev 150
```

Lo script coverage_table_convert.py è stato modificato appositamente per la creazione di questa pipeline. In particolare sono stati aggiunti i due flag: -multiplyAvg e -multiplyStdev, che servono rispettivamente per moltiplicare la media e la deviazione standard per i valori specificati. In questo caso la media viene moltiplicata per 900 e la deviazione standard per 150, queste due operazioni impattano sulla fase di binning. Infatti da varie prove empiriche abbiamo visto che il binning migliore, per i dati su cui abbiamo testato la pipeline, si ottiene settando i suddetti valori.

Prima di passare alla fase di binning, vengono eliminati gli scaffolds con media e deviazione standard pari a zero dalla coverage table

```
python3 scripts/remove_rows_tsv.py
    coverage_table.tsv
```

A questo punto della pipeline, analizzando il file coverage_table.tsv è possibile capire quali scaffolds, e quindi quali virus, sono stati trovati nei dati metagenomici.

3.3 Binning

Quest'ultimo passaggio permette la creazione di bin con lo scopo di raggruppare gli scaffolds che presentano valori molto simili all'interno della coverage table. In questo modo i bin conterranno scaffolds che appartengono a uno stesso genoma e che sono stati trovati nei dati metagenomici con una coverage molto simile. Questo tipo di binning risulta molto utile quando il numero di genomi virali completi dati in input alla pipeline è elevato. Inoltre, siccome all'interno di un bin potrebbero trovarsi scaffolds di specie di virus diversi ma "vicini" geneticamente, magari perché appartengono allo stesso genere, famiglia o ordine, il binning favorisce un'analisi dei risultati tenendo conto della classificazione tassonomica.

Il binning che è stato effettuato con due tool. Il primo tool è vRhyme, che prende in input gli scaffolds dei virus e la coverage table

```
vRhyme -i scaffoldsVirus.fasta -o resultsvRhyme/
    -c coverage_table.tsv -t 2
```

-t 2 specifica i thread da utilizzare. Purtroppo vRhyme, in tutti i casi, non ha prodotto nessun bin. Di conseguenza abbiamo deciso di usare Binsanity: un altro tool di binning che lavora utilizzando la coverage table, quindi sullo stesso input di vRhyme. Prima però di dare in input a Binsanity la coverage table, bisogna rinominarla aggiungendo .lognorm al nome e bisogna togliere l'intestazione. Per fare questo usiamo lo script:

```
python3 scripts/generate_lognorm.py
    coverage_table.tsv

```

Vale la pena sottolineare che l'input dato a Binsanity è identico all'input dato a vRhyme, lo script "generate_lognorm.py" non fa altro che cancellare la prima riga (cioè l'intestazione) della coverage table e rinominare il file. Infine eseguiamo Binsanity:

```
Binsanity -f . -l scaffoldsVirus.fasta -p -10 -c
    coverage_table.tsv.lognorm -o
    resultsBinsanity/
```

I flag -f e -l indicano la posizione e il nome del file fasta che contiene gli scaffolds da clusterizzare; -p permette di settare la precisione del binning, decrementando questo valore si ottengono bin più eterogenei, incrementandolo invece si ottengono bin omogenei. Il valore di questo flag quindi va scelto in base al dettaglio della classificazione tassonomica che si vuole ottenere e in base ai dati in input, in questo lavoro si è scelto -10 in base a varie prove empiriche. Il flag -c permette di specificare la coverage table.

4 Analisi dei risultati

In questa sezione andremo a mostrare e discutere dei virus trovati all'interno dei dati metagenomici, che hanno ottenuto un numero di scaffolds allineati maggiori di uno, e dei risultati ottenuti dal binning di BinSanity.

La pipeline è stata testata utilizzando quattro file contenenti i dati metagenomici estratti dall'intestino di pazienti affetti da IBS, per un totale di oltre 27 milioni di reads metagenomiche, e 4494 genomi completi di virus appartenenti al regno *Duplodnaviria*.

Le figure riportate nei seguenti capitoli non rappresentano la totalità degli scaffolds per questioni di spazio, ma vengono riportati solo i virus più rilevanti in termini di allineamento degli scaffolds. Per lo stesso motivo verranno analizzati solo i bin che presentano una certa classificazione.

4.1 File ERR5084065

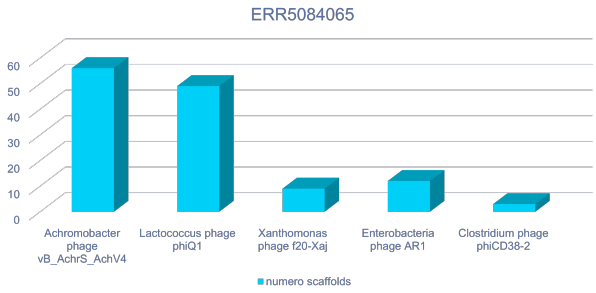

Figure 5. Virus trovati in ERR5084065

Dall'analisi della coverage table ottenuta dall'esecuzione della pipeline sul file ERR5084065, che contiene 3733279 reads, si evince che i due batteri più frequenti sono il *Achromobacter phage vB_AchrS_AchV4* con 56 scaffolds e il *Lactococcus phage phiQ1* con 49 scaffolds. Inoltre sono presenti altri tre virus batteriofagi con numero di scaffolds allineati maggiore di uno: *Xanthomonas phage f20-Xaj* con 9 scaffolds, *Enterobacteria phage AR1* con 12 scaffolds e *Clostridium phage phiCD38-2* con 3 scaffolds. Inoltre sono stati trovati 25 scaffolds appartenenti a 25 virus riconducibili a vari *Lactococcus phage* del genere *Skunavirus* differenti da *L. phage phiQ1*.

Il processo di binning ha prodotto sei bin, di seguito vengono descritti i due bin che contengono il numero di scaffolds maggiore.

Nome virus	Numero scaffolds	Lineage
Achromobacter Phage vB_AchrS_AchV4	2	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Casjensviridae; Gediminasvirus; Gediminasvirus AchV4
Lactococcus phage phiQ1	6	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Teubervirus; Teubervirus Q1
Lactococcus phage 936 group phage PhiL.6	1	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Skunavirus; Skunavirus L6
Xanthomonas phage f20-Xaj	1	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Autographiviridae; Pradovirus; Pradovirus f20
Enterobacteria phage AR1	2	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Straboviridae; Tevenvirinae; Tequatrovirus; Tequatrovirus ar1

Figure 6. Bin 30, file ERR5084065

Nome virus	Numero scaffolds	Lineage
Achromobacter Phage vB_AchrS_AchV4	6	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Casjensviridae; Gediminasvirus; Gediminasvirus AchV4
Lactococcus phage fd13	1	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Skunavirus; Skunavirus fd13
Lactococcus phage phiQ1	24	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Teubervirus; Teubervirus Q1
Lactococcus phage 16802	1	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Skunavirus; Skunavirus sv16802
Xanthomonas phage f20-Xaj	4	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Autographiviridae; Pradovirus; Pradovirus f20
Enterobacteria phage AR1	4	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Straboviridae; Tevenvirinae; Tequatrovirus; Tequatrovirus ar1
Clostridium phage phiCD38-2	2	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Leicestervirus; Leicestervirus CD382

Figure 7. Bin 60, file ERR5084065

I due bin (Figura 4 e Figura 5) raggruppano gli scaffolds che presentano media e deviazione standard, nella coverage table, molto simili. Infatti come si può osservare dalla lineage, nei bin troviamo virus che appartengono alla stessa classe (*Caudoviricetes*).

4.2 File ERR5084067

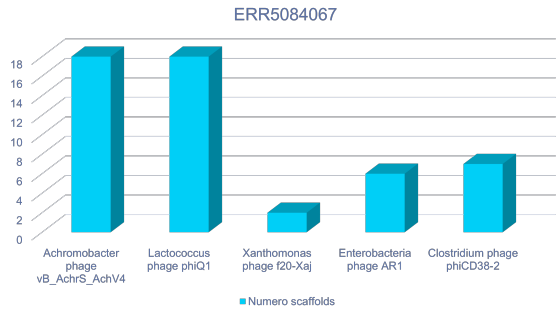

Figure 8. Virus trovati in ERR5084067

Dall'analisi della coverage table ottenuta dall'esecuzione della pipeline sul file ERR5084067,

che contiene 7184801 reads, si evince che i due batteri più frequenti sono il *Achromobacter phage B_AchrS_AchV4* con 18 scaffolds e il *Lactococcus phage phiQ1* con 18 scaffolds. Inoltre sono presenti altri tre virus batteriofagi con numero di scaffolds allineati maggiore di uno: *Xanthomonas phage f20-Xaj* con 2 scaffolds, *Enterobacteria phage AR1* con 6 scaffolds e *Clostridium phage phiCD38-2* con 7 scaffolds. Inoltre sono stati trovati 6 scaffolds appartenenti a 6 virus riconducibili a vari *Lactococcus phage* del genere *Skunavirus* differenti da *L. phage phiQ1*, e 4 scaffolds di 4 virus riconducibili alle specie *Taranisvirus taranis*, in particolare: *FP_Brigit*, *FP_Lagaffe*, *FP_Mushu* e *FP_Taranis*.

Il processo di binning ha prodotto tre bin, di seguito vengono descritti i due bin che contengono il numero di scaffolds maggiore.

Nome virus	Numero scaffolds	Lineage
Achromobacter Phage vB_AchrS_AchV4	12	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Casjensviridae; Gediminasvirus; Gediminasvirus AchV4
CrAssphage cr7_1	1	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Crassvirales; Suoliviridae; Oafivirinae; Burzaovirus; Burzaovirus coli
Lactococcus phage phiQ1	6	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Teubervirus; Teubervirus Q1
Lactococcus phage 16802	1	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Skunavirus; Skunavirus sv16802
Xanthomonas phage f20-Xaj	2	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Autographiviridae; Pradovirus; Pradovirus f20
Shigella phage SfII	1	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Myoviridae; unclassified Myoviridae
Enterobacteria phage AR1	2	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Straboviridae; Tevenvirinae; Tequatrovirus; Tequatrovirus ar1
Clostridium phage phiCD38-2	3	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Leicestervirus; Leicestervirus CD382

Figure 9. Bin 4, file ERR5084067

Nome virus	Numero scaffolds	Lineage
Achromobacter phage vB_AchrS_AchV4	3	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Casjensviridae; Gediminasvirus; Gediminasvirus AchV4
Lactococcus phage phiQ1	3	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Teubervirus; Teubervirus Q1
Enterobacteria phage AR1	1	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Straboviridae; Tevenvirinae; Tequatrovirus; Tequatrovirus ar1
Clostridium phage phiCD38-2	1	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Leicestervirus; Leicestervirus CD382

Figure 10. Bin 8, file ERR5084067

Anche in questo caso tutti i virus raggruppati appartengono alla stessa classe, cioè *Caudoviricetes*, proprio come i bin del file ERR5084065. Un'ulteriore similarità è data dalla presenza dei virus *Achromobacter phage B_AchrS_AchV4* e *Lactococcus phage phiQ1* che rappresentano i virus con numero di scaffolds maggiore nei bin.

4.3 File ERR5084069

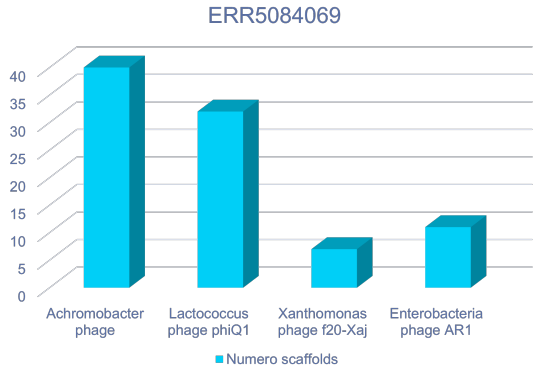

Figure 11. Virus trovati in ERR5084069

A differenza dei file ERR5084065 e ERR5084067, in questi dati non sono state rilevate tracce evidenti del virus *Clostridium phage phiCD38-2*, tuttavia è presente un alto numero di scaffolds di *Achromobacter phage B_AchrS_AchV4* (40 scaffolds) e *Lactococcus phage phiQ1* (32 scaffolds), proprio come i due file precedenti. Anche per i restanti due virus il discorso è simile.

Oltre ai virus riportati in figura sono stati trovati 25 scaffolds di 25 virus differenti riconducibili a *Lactococcus phage* differenti da *L. phage phiQ1*, proprio come nel caso di ERR5084065. Inoltre sono stati rilevati anche i virus associati a *Taranisvirus taranis*, in particolare: *FP_Mushu*, *FP_Taranis* e *FP_Toutatis*

Il processo di binning ha generato i due bin seguenti.

Nome virus	Numero scaffolds	Lineage
Achromobacter Phage vB_AchrS_AchV4	8	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Casjensviridae; Gediminasvirus; Gediminasvirus AchV4
Lactococcus phage fd13	1	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Skunavirus; Skunavirus fd13
Lactococcus phage phiQ1	11	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Teubervirus; Teubervirus Q1
Lactococcus phage 56003	1	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Skunavirus; Skunavirus sv56003
Lactococcus Phage ASCC281	1	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Skunavirus; Skunavirus ASCC281
Xanthomonas phage f20-Xaj	5	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Autographiviridae; Pradovirus; Pradovirus f20
Enterobacteria phage AR1	2	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Straboviridae; Tevenvirinae; Tequatrovirus; Tequatrovirus ar1

Figure 12. Bin 35, file ERR5084069

Anche in questo caso tutti i virus raggruppati appartengono alla stessa classe *Caudoviricetes*

Nome virus	Numero scaffolds	Lineage
Achromobacter Phage vB_AchrS_AchV4	3	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Casjensviridae; Gediminasvirus; Gediminasvirus AchV4
CrAssphage cr56_1	1	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Crassvirales; Suoliviridae; Oafivirinae; Burzaovirus; Burzaovirus faecalis
Lactococcus phage phiQ1	5	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Teubervirus; Teubervirus Q1
Lactococcus phage 13w11L	1	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Skunavirus; Skunavirus sv13w11L
Lactococcus phage 16802	1	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Skunavirus; Skunavirus sv16802
Lactococcus phage CHPC965	1	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Skunavirus; Skunavirus CHPC965
Xanthomonas phage f20-Xaj	1	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Autographiviridae; Pradovirus; Pradovirus f20

Figure 13. Bin 37, file ERR5084069

4.4 File ERR5084070

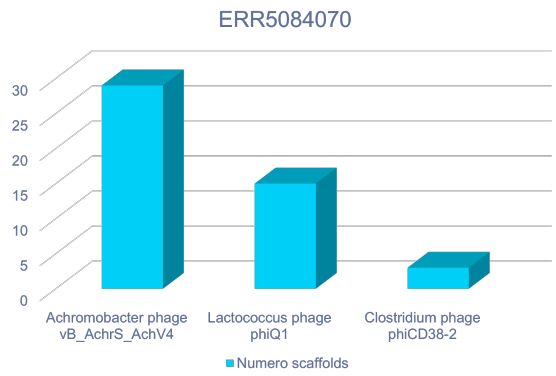

Figure 14. Virus trovati in ERR5084070

Anche in questo file i due virus predominanti sono *Achromobacter phage B_AchrS_AchV4* (29 scaffolds) e *Lactococcus phage phiQ1* (15 scaffolds), inoltre a differenza di ERR508469, è presente *Clostridium phage phiCD38-2* (3 scaffolds).

Una nota interessante, che differenzia questo file dagli altri, è l'assenza di virus *Lactococcus phage* diversi da *L. phage phQ1* Anche il risultato del binning è differente,

Nome virus	Numero scaffolds	Lineage
Achromobacter Phage vB_AchrS_AchV4	5	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Casjensviridae; Gediminasvirus; Gediminasvirus AchV4
Enterobacteria phage AR1	1	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Straboviridae; Tevenvirinae; Tequatrovirus; Tequatrovirus ar1
Clostridium phage phiCD38-2	2	Viruses; Duplodnaviria; Heunggongvirae; Uroviricota; Caudoviricetes; Leicestervirus; Leicestervirus CD382

Figure 15. Bin 44, file ERR5084070

infatti si può notare l'assenza degli scaffolds di *L. phage phiQ1*.

4.5 Risultati comuni tra i file ERR50840XX

Da quanto mostrato nelle precendenti tabelle, tutti i bin raggruppano i virus fino alla classe *Caudoviricetes*. Inoltre si può anche notare la presenza frequente di alcuni virus come: *Achromobacter phage B_AchrS_AchV4*, vari *Lactococcus phage* e *Xanthomonas phage f20-Xaj*. Infine risulta evidente che la totalità dei virus rilevati è batteriofaga.

4.6 Comparativa: Genome Detective vs Bowbin

In questa sezione si andranno a confrontare le similitudini dei genomi dei virus riscontrate da Genome Detective e Bowbin, partendo dagli stessi file che contengono i metagenomi.

4.6.1 Corrispondenze

Di seguito riportati i virus riscontrati sia in genome Detective che in Bowbin

- Lactococcus phage 936 group phage PhiB1127
- Lactococcus phage 936 group phage Phi91127
- Lactococcus lactis phage p272
- Lactococcus phage LP9207
- Skunavirus sv3R16S
- Skunavirus sv30804
- Skunavirus sv16802
- Lactococcus phage CB19
- Lactococcus phage R31
- Lactococcucs phage CB20
- Leuconostoc phage phiLN6B
- Leuconostoc phage Lmd1
- Salmonella phage vB_SenS_Sasha
- Streptococcus phage YMC-2011

La presenza di virus rilevati sia da Genome Detective che da BowBin, indicano che quest'ultima pipeline è in grado di rilevare un sottoinsieme di virus presenti nei metagenomi. Tuttavia ci sono due considerazioni da fare: nei virus rilevati da Genome Detective non è presente *Achromobacter phage vB_AchrS_ArchV4* e nemmeno *Lactoccus phage phiQ1*, riguardo al primo virus si potrebbe pensare a un falso positivo di BowBin o a un falso negativo di Genome Detective; riguardo a *L. phage phiQ1* si potrebbe pensare a un errore, o di BowBin o di Genome Detective, dovuto al numero elevato di varianti di *L. phage*. Inoltre Genome Detective ha rilevato molti virus che non appartengono al regno *Duplodnaviria*, ovviamente questi virus non sono stati segnalati da BowBin siccome non sono stati dati in input per questioni computazionali.

4.6.2 Similitudini

Di seguito verranno riportati dei confronti tra virus riscontrati da Genome Detective e Bowbin. Si avranno delle corrispondenze fino alla famiglia o classe, dato che Genome Detective avendo a disposizione un maggior numero di genomi completi dei virus, risce ad essere più dettagliato e formire più virus.

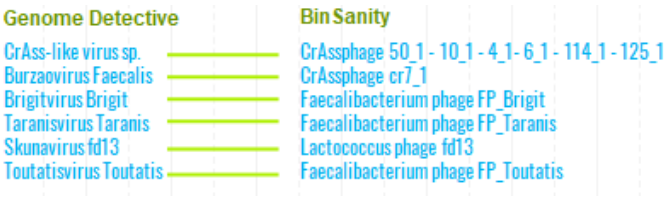

Figure 16. Similitudini tra Genome Detective e BinSanity

Il primo risultato simile che si può osservare è la presenza dei virus *CrAssphage*, Genome Detective non è riuscito a fornire le specie esatte, mentre BowBin ha identificato i seguenti: *CrAssphage 50_1*, *CrAssphage 10_1*, *CrAssphage 4_1*, *CrAssphage 6_1*, *CrAssphage 114_1* e *CrAssphage 125_1*.

La presenza degli altri risultati, in particolare dei virus *Brigitvirus brigit*, *Taranisvirus taranis* e *Toutatisvirus toutatis*, potrebbe essere dovuta alla mancata classificazione tassonomica di varianti[1]. Infatti BowBin rileva, al posto dei virus suddetti, rispettivamente *Faecalibacterium phage FP_Brigit*, *Faecalibacterium phage FP_Taranis* e *Faecalibacterium phage FP_Toutatis*. Lo stesso discorso vale per il virus *Burzaovirus faecalis* e *Skunavirus fd13*.

5 Conclusioni

La pipeline descritta in questo lavoro si è rilevata rapida nell'esecuzione dell'allineamento grazie all'indicizzazione e all'uso di Bowtie2, purtroppo però i passaggi necessari tra un tool e un altro intaccano la chiarezza del processo, quindi come lavoro futuro si potrebbe realizzare uno script per l'invocazione automatica dei vari tool e dei vari script realizzati.

Un grande punto a favore della pipeline è la scalabilità, infatti oltre all'estrazione di virus dai dati metagenomici permette l'estrazione di batteri, funghi e di qualsiasi altro regno. Questo è possibile grazie all'uso di Bowtie2, siccome è in grado di lavorare con genomi di qualsiasi organismo. Inoltre l'utilizzo di Binsanity garantisce il binning anche per organismi eucarioti, batteri e archea [11].

Purtoppo per questioni computazionali e per il non funzionamento dei tool ViruSpy e Vibrant, la pipeline è stata testata su un numero troppo piccolo di virus e tutti appartenenti al regno *Duplodnaviria*. Di conseguenza non possiamo trarre conclusioni riguardo la correlazione tra i virus che formano il viroma intestinale, l'impatto che hanno sul batterioma e la Irritable Bowel Syndrome. Inoltre, proprio perché IBS potrebbe essere causata da più cause che interagiscono tra di loro, come lavoro futuro si potrebbe testare la pipeline utilizzando un numero ben maggiore di virus e più eterogenei, e contemporaneamente testarla sui taxa di batteri e funghi che popolano l'intestino con lo scopo di scoprire legami tra questi regni.

[1] A sostegno di ciò si veda NCBI Taxonomy Browser

6 Data availability

Codice pipeline:https://github.com/strumenti-formali-per-la-bioinformatica/bowbin-pipeline-rilevazione-virus-dati-metagenomici
Dati usati per il testing della pipeline e risultati:

- Dati metagenomici:
  - ERR5084065 (https://www.ncbi.nlm.nih.gov/sra/?term=ERR5084065)
  - ERR5084067 (https://www.ncbi.nlm.nih.gov/sra/?term=ERR5084067)
  - ERR5084069 (https://www.ncbi.nlm.nih.gov/sra/?term=ERR5084069)
  - ERR5084070 (https://www.ncbi.nlm.nih.gov/sra/?term=ERR5084070)
- Genomi dei virus utilizzati (https://www.ncbi.nlm.nih.gov/labs/virus/vssi/#/virus?SeqType_s=Genome&VirusLineage_ss=Duplodnaviria,%20taxid:2731341)

Tools utilizzati:

- Samtools (https://github.com/samtools/samtools)
- vRhyme (https://github.com/AnantharamanLab/vRhyme)
- Bowtie2 (https://github.com/BenLangmead/bowtie2)
- ViruSpy (https://github.com/NCBI-Hackathons/ViruSpy)
- Vibrant (https://github.com/AnantharamanLab/VIBRANT)
- BinSanity (https://github.com/edgraham/BinSanity)
- MetaBat2 (https://bitbucket.org/berkeleylab/metabat/src/master/)

References

[1] R.M. Lovell, A.C. Ford, Clinical gastroenterology and hepatology **10**, 712 (2012)

[2] C. Canavan, J. West, T. Card, Clinical epidemiology **6**, 71 (2014)

[3] H.Y. Qin, C.W. Cheng, X.D. Tang, Z.X. Bian, World journal of gastroenterology: WJG **20**, 14126 (2014)

[4] M. El-Salhy, D. Gundersen, Nutrition journal **14**, 1 (2015)

[5] S.M. Collins, Nature reviews Gastroenterology & hepatology **11**, 497 (2014)

[6] S. Coughlan, A. Das, E. O'Herlihy, F. Shanahan, P. O'Toole, I. Jeffery, Gut Microbes **13**, 1887719 (2021)

[7] A.N. Shkoporov, C. Hill, Cell host & microbe **25**, 195 (2019)

[8] B. Langmead, S.L. Salzberg, Nature methods **9**, 357 (2012)

[9] D.D. Kang, F. Li, E. Kirton, A. Thomas, R. Egan, H. An, Z. Wang, PeerJ **7**, e7359 (2019)

[10] K. Kieft, A. Adams, R. Salamzade, L. Kalan, K. Anantharaman, Nucleic Acids Research **50**, e83 (2022)

[11] E.D. Graham, J.F. Heidelberg, B.J. Tully, PeerJ **5**, e3035 (2017)

[12] M. Vilsker, Y. Moosa, S. Nooij, V. Fonseca, Y. Ghysens, K. Dumon, R. Pauwels, L.C. Alcantara, E. Vanden Eynden, A.M. Vandamme et al., Bioinformatics **35**, 871 (2019)

[13] G. Zhao, G. Wu, E.S. Lim, L. Droit, S. Krishnamurthy, D.H. Barouch, H.W. Virgin, D. Wang, Virology **503**, 21 (2017)

[14] *fastq-join*, http://https//github.com/ExpressionAnalysis/eautils

[15] R. Schmieder, R. Edwards, Bioinformatics **27**, 863 (2011)

[16] W. Li, A. Godzik, Bioinformatics **22**, 1658 (2006)

[17] H. Li, R. Durbin, Bioinformatics **26**, 589 (2010)

[18] T. Ho, I.E. Tzanetakis, Virology **471**, 54 (2014)

[19] A.S. Waller, T. Yamada, D.M. Kristensen, J.R. Kultima, S. Sunagawa, E.V. Koonin, P. Bork, The ISME journal **8**, 1391 (2014)

[20] S.D. Ehrlich, in *Metagenomics of the human body* (Springer, 2011), pp. 307–316

[21] J. Peterson, S. Garges, M. Giovanni, P. McInnes, L. Wang, J.A. Schloss, V. Bonazzi, J.E. McEwen, K.A. Wetterstrand, C. Deal et al., Genome research **19**, 2317 (2009)

[22] J.R. Kultima, S. Sunagawa, J. Li, W. Chen, H. Chen, D.R. Mende, M. Arumugam, Q. Pan, B. Liu, J. Qin et al. (2012)

[23] D.R. Zerbino, E. Birney, Genome research **18**, 821 (2008)

[24] A.M. Bolger, M. Lohse, B. Usadel, Bioinformatics **30**, 2114 (2014)

[25] J. Brown, M. Pirrung, L.A. McCue, Bioinformatics **33**, 3137 (2017)

[26] B. Buchfink, C. Xie, D.H. Huson, Nature methods **12**, 59 (2015)

[27] *Swissprot uniref90*, https://www.uniprot.org/help/uniref

[28] N.A. O'Leary, M.W. Wright, J.R. Brister, S. Ciufo, D. Haddad, R. McVeigh, B. Rajput, B. Robbertse, B. Smith-White, D. Ako-Adjei et al., Nucleic acids research **44**, D733 (2016)

[29] A. Bankevich, S. Nurk, D. Antipov, A.A. Gurevich, M. Dvorkin, A.S. Kulikov, V.M. Lesin, S.I. Nikolenko, S. Pham, A.D. Prjibelski et al., Journal of computational biology **19**, 455 (2012)

[30] K. Deforche, BioRxiv p. 200394 (2017)

[31] M. Hunt, A. Gall, S.H. Ong, J. Brener, B. Ferns, P. Goulder, E. Nastouli, J.A. Keane, P. Kellam, T.D. Otto, Bioinformatics **31**, 2374 (2015)

[32] H.H. Lin, Y.C. Liao, Gigascience **6**, gix003 (2017)

[33] A. Shariati, F. Fallah, A. Pormohammad, A. Taghipour, H. Safari, A.S. Chirani, S. Sabour, M. Alizadeh-Sani, T. Azimi, Journal of cellular physiology **234**, 8550 (2019)

[34] G.M. Boratyn, J. Thierry-Mieg, D. Thierry-Mieg, B. Busby, T.L. Madden, BMC bioinformatics **20**, 1 (2019)

[35] *Glimmer*, http://ccb.jhu.edu/software/glimmer/index.shtml

[36] D. Li, C.M. Liu, R. Luo, K. Sadakane, T.W. Lam, Bioinformatics **31**, 1674 (2015)

[37] *Viruspy: a pipeline for viral identification from metagenomic samples*, https://github.com/NCBI-Hackathons/ViruSpy

[38] K. Kieft, Z. Zhou, K. Anantharaman, Microbiome **8**, 1 (2020)

[39] M.H. Ansari, M. Ebrahimi, M.R. Fattahi, M.G. Gardner, A.R. Safarpour, M.A. Faghihi, K.B. Lankarani, BMC microbiology **20**, 1 (2020)