Anàlisis de l'evolució del genoma de *Physcomitrella patens*

Pol Vendrell Mir
Màster en Bioinformàtica i Bioestadística
Anàlisis de dades òmiques

Josep Mª Casacuberta Suñer
Roser Pratdesaba Moreno

24 de maig del 2017

FITXA DEL TREBALL FINAL

Títol del treball:	Anàlisis de l'evolució del genoma de *Physcomitrella patens*
Nom de l'autor:	*Pol Vendrell Mir*
Nom del consultor/a:	*Josep Maria Casacuberta Suñer*
Nom del PRA:	*Roser Pratdesaba Moreno*
Data de lliurament (mm/aaaa):	*05/2017*
Titulació o programa:	*Màster en Bioinformàtica i Bioestadística*
Àrea del Treball Final:	*Anàlisis de dades òmiques*
Idioma del treball:	*Català*
Paraules clau	*Evolució, elements transposables, polimorfismes*

Resum del Treball (màxim 250 paraules): *Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball*

En aquest treball s'ha analitzat l'evolució de *Physcomitrella patens*. Es realitzen dues aproximacions: per una part s'ha estudiat els elements transposables polimòrfics en diferents ecotips de l'organisme a partir de dades de *Next Generation Sequencing.* Per altra banda, s'ha estudiat l'evolució del genoma de *P. patens* en un clon reproduït vegetativament durant els darrers 8 anys. Aquest clon prové de l'utilitzat per realitzar l'anotació publicada l'any 2008, el genoma de referència. Això s'ha realitzat gràcies a la disposició de diferents reseqüenciacions en anys diferents. Totes aquestes anàlisis s'han realitzat utilitzant programes de detecció de polimorfismes de transposons que permeten 'anàlisis de seqüenciacions en format *paired-end.*

A més a més s'ha descrit quina és la millor metodologia per tal de detectar els polimorfismes de transposons i s'ha obtingut els resultats pertinents dels polimorfismes en diferents ecotips anomenats *Villersexel* i *Reute*.

S'ha comprovat que els elements transposables tipus *RLG1* són els que més moviment presenten en el genoma de *Physcomitrella patens* i que aquests tenen tendència a inserir-se dintre d'altres gens.

També s'ha comprovat que l'ecotip *Reute* pels polimorfismes obtinguts és molt més proper a l'ecotip de referència anomenat *Gransden* mentre que en l'ecotip *Villersexel* presenta molts més polimorfismes i per tant és més distant a la referència.

Finalment, s'ha comprovat que amb les dades disponibles de les seqüenciacions durant els darrers 8 anys no ha estat possible detectar polimorfismes de transposons.

Abstract (in English, 250 words or less):

In this paper, we analyzed the *Physcomitrella patens* evolution. We carried out two approximations to study this evolution.

On the one side, we analyzed the polymorphisms in two ecotypes: *Villersexel* and *Reute* through data obtained by Next Generation Sequencing.

On the other side, we analyzed the evolution of *P. patens* through multiple resequencing from the original clone used to generate the annotation that had been vegetative propagated during the last 8 years. This has been possible due to some resequencing samples that come from other ones resequenced during that time.

We examined these data using programs that detect polymorphism transposable elements which allow the analysis of sequencing using paired-end.

We described the best methodology in order to detect polymorphism transposable elements and we obtained the results from the polymorphic elements in the ecotypes *Villersexel* and *Reute*.

We verified that the *RLG1* elements are the ones that present most movement in the *Physcomitrella patens* genome and that these elements tend to get inserted inside other transposable elements.

Moreover, we confirmed that the *Reute* ecotype is highly similar to the *Gransden* ecotype (reference genome) and at the same time the *Villersexel* presents much more polymorphism, which means it is more distant to the reference.

Finally, we were able to verify that with the available samples from the last 8 years it has not been possible to detect any polymorphic transposable element.

Índex

1. Introducció

1.1 Context i justificació del Treball

Durant els darrers anys hem viscut com s'han incrementat de forma molt significativa la quantitat de genomes seqüenciats. S'han publicat, fins a dia d'avui, uns 24000 genomes aproximadament de gran diversitat d'espècies (https://www.ncbi.nlm.nih.gov/genome/browse/ [21/05/2017]). Entre ells trobem publicats genomes d'organismes models com *E. coli (*Blattner *et al*., 1997) en el cas de bacteris, *D. Melanongaster* (Adams *et al*., 2000) en animals invertebrats, *Mus musculus (Gregory et al.,* 2002) en el cas de mamífers *o A. thaliana* (*Arabidopsis* genome initiative*, et al.,* 2000) en el cas de plantes. Aquests organismes s'han seleccionat en general per la seva facilitat de manipulació, per tenir un genoma petit o bé per un alt grau de coneixement de les seves rutes metabòliques o fisiologia d'aquests organismes entre altres causes.

Entre aquests organismes model trobem *Physcomitrella patens* (Rensing *et al*, 2008), una planta briòfita que presenta tot un seguit de facilitats:

- És fàcil de propagar i créixer a partir de cèl·lules somàtiques
- En la majoria del seu cicle vital és un organisme haploide, fet que facilita la seva manipulació i estudi dels seus gens (essent més fàcil d'obtenir fenotips observables de les mutacions).
- Presenta una alta freqüència de reparació del seu DNA per recombinació homòloga, cosa que permet generar mutacions dirigides de gens i fer genètica reversa.

La seqüenciació i anotació del genoma de *Physcomitrella patens* es va publicar l'any 2008 (Rensing *et al*, 2008), seqüenciant a partir del teixit protonema, teixit és haploide. *Physcomitrella patens* pertany al regne de les plantes a la divisió dels briòfits de l'ordre de les funarials.
El genoma d'aquest organisme es divideix en 27 cromosomes amb una longitud total de 47 mil milions de bases nucleotídiques. Aquest presenta tot un seguit de característiques que el fan de particular interès:

- Presenta una heterocromatina difosa al llarg de tots els seus cromosomes. Normalment es sol trobar la heterocromatina propera a la zona pericentromèrica, no és el cas per *P. patens.*
- Un 57% del genoma està format per elements transposables, on gran part d'aquestes zones, que contenen els elements transposables, coincideixen amb les regions d'heterocromatina (Rensing *et al.*, 2017).
- La gran majoria dels elements transposables estan formats per elements de tipus *Gypsy* de la família RLG1, que ocupa el 25% del genoma.

Això fa que sigui un genoma de particular interès per l'estudi de l'impacte dels transposons sobre l'heterocromatina així com en l'evolució dels propis genomes (Rensing *et al.*, 2017).

Els transposons han jugat un paper important en l'evolució de tots els genomes (Lisch, 2013). La dinàmica de transposició és molt diferent entre diferents organismes; així com en humans els elements més freqüents són els de tipus *LINES* (Nekrutenko *et al.,* 2001). En plantes els elements transposables més freqüents són els de tipus retrotransposó, així com els de tipus *LTR i MITES* (Lisch, 2013).

Les insercions i/o delecions de transposons properes a gens poden donar una gran varietat de fenotips en diferents espècies. En plantes, alguns dels casos més evidents són la inserció d'un transposó proper al gen *Vvmby1A* responsable de produir el color negre característic del raïm (Lisch, 2013). També trobem casos similars en tomàquet (Lisch, 2013) o en blat de moro (Lisch, 2013)

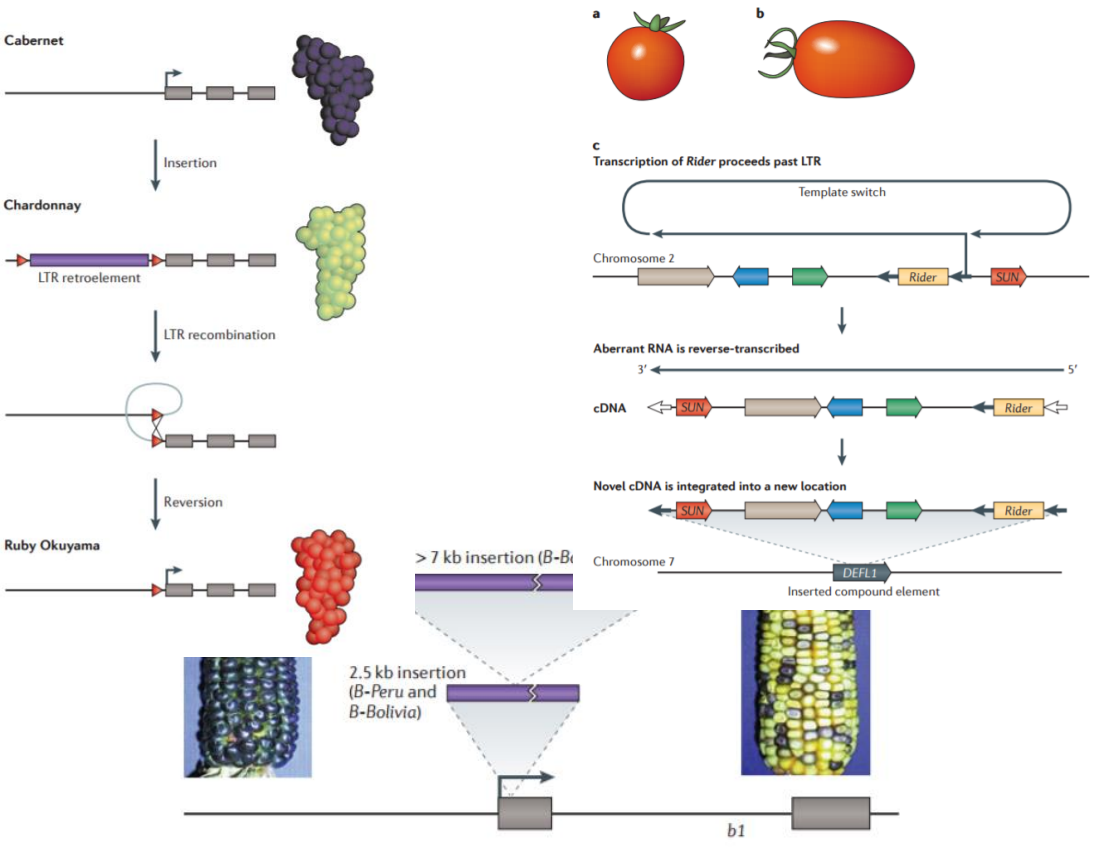

Figura 1: Efecte dels polimorfismes de transposons en diferents espècies vegetals. A dalt a l'esquerra tenim com la inserció d'un retrotransposó causa el silenciament total del gen *Vvmby1A* causant el color del raïm blanc o com es pot produir altres fenotips per delecions parcials del transposó com és el color vermell/ porpra de certes varietats de raïm. A la dreta com l'efecte d'un retrotransposó causa canvis estructurals en els gens *DEFL1* causant el fenotip de tipus tomàquet pera o com insercions de determinats elements causen problemes de pigmentació al blat de moro.(font: Lisch, 2013).

L'organisme model de *Physcomitrella patens* que es va seqüenciar i anotar l'any 2008 pertanyia a l'ecotip *Gransden* (Rensing *et al.*, 2008). Des de llavors es van propagar vegetativament diferents clons de l'organisme seqüenciat entre diferents laboratoris disposant de mostres procedents d'aquell clon.

El clon es propaga de forma vegetativa, fragmentant el seu teixit amb un disruptor durant la fase de protonema i repicant-lo cada dues setmanes, en els medis de cultiu adequats; únicament, un cop a l'any aproximadament, entra en cicle sexual formant espores i tornant a regenerar el teixit a partir d'aquestes espores. El cicle reproductiu de la molsa el podem trobar a la següent imatge:

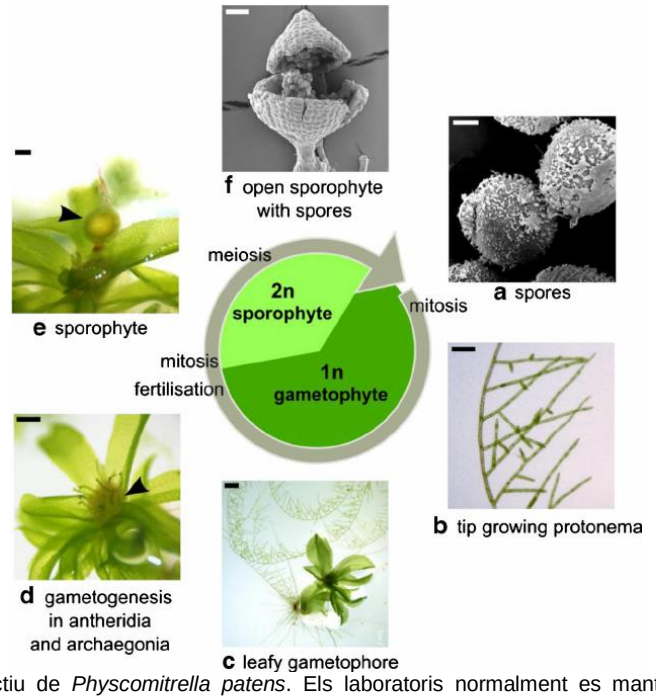

Figura 2: Cicle reproductiu de *Physcomitrella patens*. Els laboratoris normalment es manté entre les fases b i c fragmentant-se per tal de mantenir-lo en el mateix estadis de protonema i gametofor sense generar esporòfits. Font: Müller *et al.,* 2016).

Hi ha varis grups que treballen utilitzant *Physcomitrella patens* com a organisme model per provar eines de biologia molecular (Rensing *et al*., 2017). Entre aquests grups es troba el grup *DNA repair and genome engineering* al INRA, concretament Centre de Versailles-Grignon. Aquest grup manté clons de *Physcomitrella patens* provinents del que es va seqüenciar l'any 2008 (Rensing *et al*., 2008). Utilitzen l'organisme *Physcomitrella patens* per realitzar estudis dels mecanismes de reparació del DNA cel·lular. Entre aquests, estudien l'activitat *off-target* de Meganucleases així com del sistema *CRISPR-Cas9*. Per tal de realitzar aquests estudis, van seqüenciar mitjançant mètodes de *Next generation sequencing (NGS) paired-end* diferents clons no tractats amb meganucleases com a controls així com clons tractats tant amb meganucleases

com amb el sistema *CRISPR-Cas 9* per comprovar la possible activitat *off target* d'aquestes nucleases.

L'anàlisi de les diferents mostres controls (sense estar tractades per sistemes de tall dirigit) tenien una gran quantitat de *SNP*s comparats amb la seqüencia de referència (Rensing *et al*., 2008). Per corroborar les dades van tornar a cultivar clons del any 2010 sense tractar amb nucleases i van seqüenciar-les de nou. Van observar que entre les mostres del 2016 i 2010 hi havia *SNP*s que no eren presents en les mostres originals així com amb les mostres del 2016 hi ha mutacions que no eren presents ni el 2010 ni el 2008.

Així doncs, les anàlisi de *SNPs* mostren que el genoma del clon de *P.patens* amb el que treballem al laboratori ha evolucionat des del moment en que va ser seqüenciat l'any 2008. És per això que en aquest treball ens hem plantejat si els *TEs* podrien haver modificat el genoma *de P.patens* durant aquest període.

Els transposons, a diferència dels *SNP*s, presenten un seguit de característiques que els fa molt diferents d'aquests polimorfismes:

Per una banda els efectes que produeixen sobre el genoma són importants. Els polimorfismes produeixen moviments de grans quantitats de seqüència (en el cas de retrotransposons entre 6 i 8 kb); aquests poden alterar l'expressió de gens propers i fins i tot produir canvis estructurals en el genoma; comparats amb altres tipus de polimorfismes. També els transposons presenten una dinàmica variable al llarg del temps, podent passar gran quantitat de temps sense que es produeixin polimorfismes i, en canvi, durant períodes breus expressar-se fortament i produir moltes noves insercions i/o delecions (Ewing, 2015).

Difereixen en aquesta característica amb els *SNP*s que mantenen una dinàmica molt més constant al llarg del temps (Stoneking, 2001). Finalment, els transposons també presenten el desafiament que en contrast amb els *SNP*s han estat poc estudiats mitjançant tècniques de *NGS*. Això és degut a la pròpia naturalesa dels transposons que dificulten el seu estudi. Fins que no han sorgit noves tècniques que permeten la seva detecció en dades de *High throughput,* els estudis fets a nivells de seqüenciacions no han estat molt nombroses (Ewing, 2015). Tots aquests motius porten a l'estudi d'aquests ecotips de *Physcomitrella patens.*

A més a més, amb la contribució de les dades facilitades pel grup de *Versailles-Grignon (DNA repair and genome engineering)* es pot estudiar si hi ha hagut canvis estructurals en el genoma de *Physcomitrella patens* degut a la transposició durant aquest període de temps i si això pot implicar nous fenotips en el genoma de referència. També ens dóna informació parcial sobre si, com passa en altres organismes models mantinguts al laboratori (Sniegowski *et al.,* 1997), a mesura que van succeint noves generacions es produeixen canvis estructurals al genoma i en quina freqüència això succeeix.

S’estudia, en aquest treball, a partir de *raw data* de diferents reseqüenciacions realitzades durant els darrers anys sobre clons propagats vegetativament a partir del clon seqüenciat i anotat l’any 2008. A més a més, s’estudien els polimorfismes en diferents ecotips. Per altra banda, es defineix un *workflow* per tal d’optimitzar la cerca de transposons polimòrfics. Permetent així aplicar el contingut del màster a un cas real i d’interès. Finalment, també s’apliquen nous programes per a l’estudi de genomes no estudiats durant el màster i que han anat sorgint durant els darrers anys.

1.2 Objectius del Treball

Objectius generals

1. Avaluar les eines bioinformàtiques per la detecció de polimorfismes generats pel moviment de transposons.

2. Estudiar l'impacte dels transposons en el genoma de *Physcomitrella patens* i l'evolució d'aquest durant un període breu de temps.

3. Detectar els polimorfismes causats per elements transposables en diferents ecotips de *Physcomitrella patens*.

Objectius específics

1. Analitzar quin és el programa més adequat per tal d'alinear les seqüències contra el genoma i recuperar el màxim de *discordands reads* i de millor qualitat.
2. Comparar els diferents programes d'anàlisis d'insercions de transposons en el genoma.
3. Establir un procés per tal d'analitzar els polimorfismes de transposons en les seqüències obtingudes.
4. Determinar la quantitat de polimorfismes causats per transposons (tant insercions com delecions) que han aparegut durant el període de temps comprès en l'estudi.
5. Establir en conjunt els polimorfismes presents en les reseqüenciacions, agrupant-les per clon o bé per any, en contrast d'individualment.
6. Comprovar l'efecte de la creació de *pools* sobre les dades, comparant diferents agrupacions de reseqüenciacions: Bé les dades en brut (sense filtrar), les dades filtrades o els *bam files* obtinguts.
7. Avaluar l'evolució del genoma de *Physcomitrella patens* durant el període de temps comprès entre 2008 i 2016 gràcies a l'activitat dels seus transposons.
8. Analitzar l'impacte de les transposicions respecte els gens propers.
9. Comparar amb les seqüenciacions de diferents ecotips per detectar els artefactes.
10. Analitzar els polimorfismes causats per transposons en diferents ecotips.
11. Realitzar el procés d'anotació de transposons en diferents ecotips.

1.3 Enfocament i mètode seguit

Les dues possibles estratègies pel desenvolupament del treball són:

- La primera opció més senzilla i més directe seria mapejar, utilitzant *BWA mem*, el conjunt de *reads* i a continuació analitzar aquest arxiu amb qualsevol dels programes de detecció de polimorfismes de transposons com *jitterbug o pindel*. A partir de les dades obtingudes valorar la opció de crear *pools* en funció dels polimorfismes detectats així com del *coverage* del genoma. En cas de que aquest fos molt baix es crearien *pools* i es repetiria l'anàlisi També caldria mirar les delecions que s'hagin produït amb *pindel* i finalment establir la comparació amb el nombre de polimorfismes fent una anàlisis estadística amb R.

  Aquesta guia presenta tot un seguit de problemes: En primer lloc sinó es contrasta els diferents processos de filtratge pot ser que s'eliminin *reads* essencials per la detecció de transposons com poden ser els *discordance reads* (on en una seqüenciació *paired-end* un dels dos *reads* mapeja a un altra lloc del genoma). El segon problema és que no s'analitzen tampoc el funcionament dels programes abans de fer les anàlisis, aquest fet pot comportar que el mètode triat no sigui el més adequat. Es considera que aquesta opció no és la més adequada.

- La segona opció és més conservadora i requereix una quantitat de temps superior però a part d'enriquir l'estudi s'analitzen amb més rigor les dades per tant molt probablement tindrà més validesa per a l'estudi. A continuació es detallen els diferents passos seguits per tal de realitzar aquest estudi:

Primerament es comprova quin és el procés de filtratge més adient per als *reads* de mala qualitat conservant el format *paired-end*. En segon lloc també es compara quin és el millor procés de mapeig per tal de recuperar el màxim nombre de *reads* de qualitat i mapejats correctament així com *discordand reads* de qualitat que ens permetin la detecció de transposons. No obstant, vista la gran quantitat de programes de detecció d'insercions que s'ha anat desenvolupant en els darrers tres anys (Ewing, 2015) és difícil seleccionar a cegues un programa per tal de dur a terme l'estudi. Per això, primerament s'intenta repetir l'estudi realitzat en l'estudi de Rishishwar *et al. (*2016). En aquest estudi es compararen diferents mètodes de detecció de polimorfismes en funció de la seva precisió i sensibilitat amb aquells programes que no s'han comparat el seu funcionament amb altres programes. Un cop seleccionat el mètode per detectar polimorfismes es realitza l'estudi de la detecció dels polimorfismes. Analitzant primerament les dades, comprovant el *coverage* d'aquestes dades i si és necessari la creació de *pools* d'aquestes. Es comprova quin és el millor mètode per desenvolupar *pools;* si agrupant els *BAM files* o bé *els fastq* i com alteren els resultats. Un cop realitzat aquesta anàlisis i haver obtingut els polimorfismes causats per insercions i/o delecions es comprova

aquests polimorfismes experimentalment, mitjançant *PCR* i seqüenciació d'alguns d'aquests polimorfismes detectats

Per tots aquests motius es considera que la segona opció és la més adequada. Ja que amb un temps breu es pot assolir els objectius filtrant correctament les dades i utilitzant un programari idoni pel cas. Finalment, tota aquesta informació quedarà reflectida tant en la memòria com en la presentació.

1.4 Planificació del Treball

Tasques

El treball s'ha dividit en les següents tasques:

- Realitzar el procés de *trimming* de les seqüències, seleccionar el mètode de *trimming* en funció d'aquell que s'obtingui les millors estadístiques en relació a qualitat.
- Realitzar el procés d'alineament de les reseqüenciacions contra el genoma de referència, de nou comparar entre dos mètodes d'alineament (*BWA aln i BWA mem*) aquell en què es recuperin més seqüències de més qualitat i també en aquell procés que es recuperin una major quantitat de *discordant reads.*
- Partint de l'article (Rishishwar *et al.,* 2016), repetir les anàlisis afegint els programes que han anat sorgint des de llavors (Ewing, 2015) o que no han estat inclosos. En funció dels resultats obtinguts seleccionar el millor mètode per la detecció dels transposons.
- A partir del mètode seleccionat cercar els diferents polimorfismes d'inserció per transposons a les seqüències i filtrar-les per qualitat.
- Detectar els polimorfismes de deleció utilitzant *pindel*, seleccionar i filtrar aquelles delacions detectades relacionades amb transposons.
- Valorar si és necessari agrupar les dades o bé per clons o agrupar-les per anys i repetir els dos passos anteriors.
- En cas que s'hagin creats *pools* comprovar amb alguna mostra si agrupar les dades entre els *fastq* o els *bam* altera el resultat en el procés de detecció de polimorfismes.
- Comparar entre els diferents períodes temporals: comprovar si les insercions i delecions visualitzades el 2010 són presents el 2016 i si es pot evidenciar una evolució durant aquest període.
- Comprovar si els polimorfismes detectats poden estar afectant gens propers causant uns possibles efectes fenotípics.
- Comparar amb les seqüenciacions de diferents ecotips per detectar els artefactes.
- Analitzar l'ecotip *Reute* i *Villersexel* per tal d'analitzar l'evolució durant aquest període de temps.
- Finalment, comparar els resultats obtinguts amb la quantitat i distribució de *SNP*s i els polimorfismes d'inserció.
- Verificar els resultats obtinguts mitjançant diferents *PCR*s dels polimorfismes

Calendari

La planificació del treball ha estat la següent:

Nombre	Duración	Inicio	Fin
Trimming de les seqüències, contrast i selecció del mètode	6d?	14/03/2017	19/03/2017
Alineament de les seqüències contra el genoma de referència	3d?	20/03/2017	22/03/2017
Anàlisis dels polimorfismes, creació de pools i repetició del procés	12d?	23/03/2017	03/04/2017
Comparar els polimorfismes d'aquest període de temps amb ecotip	5d?	04/04/2017	08/04/2017
Filtratge dels polimorfismes	1d?	09/04/2017	09/04/2017
Comparar diferents programes de detecció d'insercions	5d?	10/04/2017	14/04/2017
Comprovació dels polimorfismes	3d?	15/04/2017	17/04/2017
Finalització de l'anàlisis de dades	3d?	18/04/2017	20/04/2017
Elaboració de la memòria	53d?	13/03/2017	04/05/2017
Correcció i revisió de la memòria	6d?	05/05/2017	10/05/2017
Validació dels resultats experimentalment	9d?	27/04/2017	05/05/2017
Elaboració de la presentació	8d?	11/05/2017	18/05/2017
FInalització de les correccions de la presentació i l'article, entrega	4d?	19/05/2017	22/05/2017

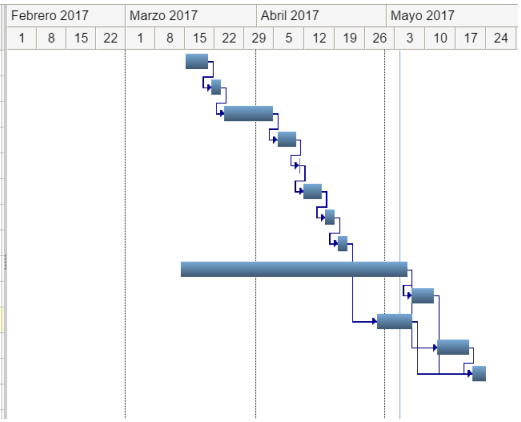

Figura 3: Planificació temporal del treball realitzat

Fites

Les fites amb el seu temps corresponent estan descrites a continuació:

-Realitzar el procés de *trimming* de les seqüències, seleccionar el mètode de *trimming* en funció d'aquell que s'obtingui les millors estadístiques en relació a la qualitat: 19 de març del 2017

- Realitzar el procés d'alineament de les reseqüenciacions contra el genoma de referència: 22 de març del 2017

- Comprovar diferents mètodes de detecció de polimorfismes de transposons i seleccionar el mètode més adequat: 2 d'abril del 2017

- Anàlisis dels polimorfismes de transposons, valoració de si cal crear pools de les dades i repetició de les dades en cas de necessitat: 9 d'abril del 2017

- Contrast dels polimorfismes entre els diferents períodes de temps: 12 d'abril del 2017

- Finalització de la comparació dels polimorfismes de transposons amb els *SNP*s 17 d'abril del 2017

- Finalització de l'anàlisis de dades 20 d'abril del 2017

-Finalització de la memòria escrita: 4 de maig del 2017

- Finalització de les correccions i revisió de la memòria: 10 de maig del 2017

-Finalització de la elaboració de la presentació 18 de maig del 2017

- Finalització de les correccions i revisió de la presentació, entrega de la revisió: 22 de maig del 2017

- Data límit de l'entrega de la presentació: 24 de maig del 2017

-Defensa pública del treball del 7 de juny del 2017 al 21 de juny del 2017

Anàlisis de riscos

- Coincidència de treballs i PECs en el desenvolupament del TFM: Durant les setmanes del 17/04/2017 al 30/04/2017 coincideix amb l'elaboració d'una PEC de l'assignatura anàlisis i regressió així com del 6 d'abril al 25 d'abril hi ha l'elaboració d'una PEC d'anàlisis òmiques. En aquests cas òbviament caldrà destinar un nombre elevat d'hores també a la elaboració dels treballs. Per tal de solventar això s'intentarà resoldre les PECs al més aviat millor així com destinar més hores diàries a la elaboració d'aquestes PEC per tal que no repercuteixin en la elaboració del TFM.

- Coincidència amb les pràctiques curriculars: En determinats moments pot ser que el volum de feina a les pràctiques s'incrementen, però hi ha una alta flexibilitat laboral. En tot cas s'intentarà incrementar el nombre d'hores durant el més d'abril destinades a la elaboració de les PEC així com el TFM a costa de les pràctiques, que es recuperaran posteriorment a finals de maig i inicis de juny.

-Estudi de *SNP*s: L'estudi de *SNP*s es realitza en paral·lel per un altre laboratori i pot ésser que aquest no arribi en el termini de temps marcat i per tant no es pugui comparar a temps en la elaboració de la memòria.

1.5 Breu sumari de productes obtinguts

A partir de la realització d'aquest treball s'espera obtenir els següents resultats:

- Memòria: A la memòria es descriuen tots els processos per tal d'arribar als productes finals. En aquesta memòria també es descriuen totes les tasques fetes i citades a la seva corresponent bibliografia. S'adjunta a més a més, taules i figures dels processos realitzats així com de les dades obtingudes. A cadascun dels apartats d'aquest treball es descriu la metodologia seleccionada i justificada, els resultats obtinguts seguint aquesta metodologia, així com les conclusions a les que es poden arribar.

- Presentació: Es presenten a mode de resum, la necessitat per la qual s'ha realitzat aquest treball, els diferents resultats obtinguts i un resum del treball fet així com les conclusions.

- Taula de polimorfismes durant el període 2008-2016: Llistat on es descriuen tots els polimorfismes descrits durant aquest període de temps.

- Llistat de polimorfismes: En format *gff3.* En aquest format es descriu posició dels polimorfismes al genoma, si són insercions o delecions respecte la seqüència de referència. Així també s'especifica els atributs descrits per els programes de detecció de polimorfismes. S'han realitzat varis arxius per cadascun dels ecotips així com per els diferents períodes de temps.

1.6 Breu descripció dels altres capítols de la memòria

El treball es divideix en tres parts:

Un primer bloc destinat al procés de detecció de polimorfismes mitjançant eines bioinformàtiques. Aquí es definirà un possible *Workflow* per la detecció de polimorfismes, així com quines alternatives hi ha a dia d'avui tant per dur a terme la detecció de polimorfismes de transposons com per preparar els arxius per tal de poder dur a terme la detecció.

El segon bloc està destinat a la detecció de polimorfismes en diferents ecotips de *P. patens,* per una part en l'ecotip *Villersexel* i per altra part en l'ecotip *Reute.*

En el darrer bloc s'estudien els polimorfismes produïts en el genoma de referència de *P. patens* corresponents a l'ecotip *Gransden* entre l'any 2008 i l'any 2016. Els capítols queden descrits en els següents apartats:

2.- Procés de detecció de polimorfismes d'elements transposables

2.1.-Introducció
On es descriu el procés d'anàlisi que es durà a terme, la necessitat de dur a terme aquest procés, així com els possibles *workflow* que es poden utilitzar per resoldre el problema plantejat.

2.2.- Selecció del mètode de filtratge de les dades (*trimming)* per qualitat
En aquest capítol es descriuen les alternatives alhora de filtrar les dades en brut i quina metodologia s'ha escollit en aquest cas per dur a terme l'estudi.

2.3.-Alineament de les seqüenciacions contra el genoma: *BWA Aln* vs *BWA* mem
Comparació de dos mètodes d'alineament de dades contra el genoma mitjançant diferents algorismes, les principals avantatges i desavantatges que presenta cada mètode i selecció de quin s'utilitza.

2.4.-Programes de detecció de polimorfismes: Comparació de programes i selecció
Selecció dels programes per detectar les insercions i delecions polimòrfiques en els diferents genomes.

2.5.-Metodologia escollida
A partir de tota la informació generada selecció de la metodologia per tal de detectar els polimorfismes de transposons.

3.- Detecció de polimorfismes del genoma de *P. patens* en diferents ecotips

3.1.- Estratègia seguida per la detecció de polimorfismes

Estratègia utilitzada per a detectar els polimorfismes de transposons entre els diferents ecotips.

3.2.-Millora de la detecció d'elements transposables en l'ecotip *Villersexel*

Detecció de polimorfismes de transposons dintre d'altres transposons en l'ecotip *Villersexel*. Comparació amb els resultats previs i resum dels resultats obtinguts.

3.3.-Validació de l'anotació en l'ecotip *Villersexel*

Comprovació dels resultats obtinguts mitjançant validació experimental, disseny de *primers* i comprovació mitjançant *PCR*.

3.4.-Detecció de polimorfismes de transposons en l'ecotip *Reute*

Detecció de polimorfismes de l'ecotip en *Reute* i resum dels resultats obtinguts.

4.- Anàlisis de l'evolució del genoma de *P. patens* durant els darrers 8 anys

4.1.- Introducció

Plantejament del problema, què es el que es va voler comprovar en aquest estudi i com es va realitzar.

4.2.- Detecció de polimorfismes en els diferents genomes

Detecció dels polimorfismes en les diferents seqüenciacions realitzades durant els diferents períodes de temps. Efectes de la creació de *pools* de les dades i anàlisis dels diferents polimorfismes.

4.3.- Detecció d'artefactes en les seqüències, validació de les dades

Detecció dels artefactes en les dades disponibles per comparació amb dades disponibles d'ecotips.

4.4.- Resum dels resultats obtinguts

Resum de tots els resultats obtinguts així com de les conclusions a les quals s'ha arribat durant el desenvolupament d'aquest apartat del treball.

4.5.- Validació experimental dels resultats

A partir dels resultats obtinguts validació mitjançant *PCR.*

2.- Procés de detecció de polimorfismes d'elements transposables

2.1.-Introducció

Els elements transposables tenen una gran rellevància i impacte en els genomes. Molts factors de transcripció en plantes i animals deriven directament de transposons, així com promotors o bé llocs d'unió de factors de transcripció. En la majoria de casos però, l'impacte més obvi dels elements transposables són les mutacions causades per la seva mobilitat, donant lloc a tot un seguit de polimorfismes al llarg del genoma. Aquests poden generar tant guanys o pèrdues de funcions de gens, reorganitzacions gèniques així com silenciament de gens, entre altres efectes (Lisch, 2013). Els elements transposables són d'especial interès per a la generació de variabilitat entre les diferents espècies o varietats i són una eina important per a l'estudi de l'evolució tant de varietats silvestres com d'interès agroalimentari (Lisch, 2013).

Els darrers estudis evolutius realitzats en el camp de les dades òmiques destaquen la importància de l'estudi dels polimorfismes de transposons conjuntament amb els estudis de *SNP*s, petits *indels* i canvis estructurals del genoma (altres insercions, delecions o inversions causades per altres fenòmens no relacionats amb la transposició) (Shendure *et al*., 2008).

Tot i que l'estudi dels polimorfismes causats per *SNP*s en dades de *NGS* està àmpliament estudiat i hi ha una gran quantitat d'eines desenvolupades (Nielsen *et al*., 2011), no succeeix el mateix pels polimorfismes causats per transposons en què solament durant els darrers anys s'han desenvolupat una gran quantitat d'eines per estudiar els polimorfismes de transposons a partir de dades de *NGS* (Ewing, 2015).

Les dinàmiques dels transposons són molt diferents als *SNP*s. Mentre que els *SNP*s solen acumular-se al llarg del temps sent més fàcil datar quan s'han produït aquestes mutacions, la dinàmica de transposició sol ser molt variable al llarg del temps podent alternar entre períodes en què hi hagi una gran quantitat de transposició i altres en què els esdeveniments de transposició succeeixin en molt menor mesura (Feschotte *et al.,* 2002).

A més a més, aquests esdeveniments de transposició solen tenir un impacte molt major el que pot causar un sol SNP. Són elements d'una mida considerable que al transposar poden alterar completament l'expressió gènica o l'estructura del genoma, per exemple (Lisch, 2013).

La detecció dels polimorfismes de transposons es pot dur a terme mitjançant varies aproximacions; bé mitjançant tècniques de biologia molecular com *Sequence-specific amplification polymorphisms (SSAP)* que es basa en fragmentar el genoma de forma aleatòria i amplificar mitjançant *PCR* els fragments que contenen insercions recents(Porceddu, *et al.,* 2002 o bé mitjançant tècniques de seqüenciació o *next generation sequencing (NGS)* (Ewing, 2015) En aquest treball ens centrarem en aquesta darrera tècnica.

Gràcies a la proliferació de grans quantitats de dades generades durant els darrers anys mitjançant *NGS* és possible analitzar aquests polimorfismes de transposons mitjançant varies aproximacions. Una d'aquestes aproximacions i la que s'utilitzarà en aquest treball es basa en la utilització de seqüenciacions *paired-end* (Gilly *et al.*, 2014). En aquestes, es seqüencien fragments d'ADN d'una longitud de 100-120 pb en parelles separades per una longitud fixada entre ells, normalment d'uns 400-500 pb. Aquestes seqüenciacions es realitzen normalment amb la tecnologia *illumina sequence (*Bennet*, 2004).* Coneixent la distància que separa cada parella de *reads,* els algoritmes d'aparellament permeten alinear els *reads* al genoma, fins i tot en zones altament repetitives. Quan un d'aquests dos *reads* alinea a un altre punt del genoma i l'altra alinea en la posició pertinent parlem de que hi ha un *discordand read* (Gilly *et al.*, 2014) (situació que es produeix quan no es troba el *read a la posició esperada).*

Els programes de detecció de polimorfismes de transposons (Ewing, 2015) utilitzen aquesta informació proporcionada pels alineaments contra el genoma per realitzar la detecció dels polimorfismes.

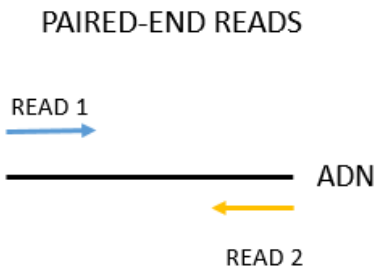

Figura 4: seqüenciació de DNA paired-end on es seqüencien fragments d'ADN de determinada mida (uns 100pb) iniciant pels seus extrems

Bàsicament hi ha dos tipus de polimorfismes de transposons:

- Per una part trobem les **delecions (**Ye *et al.,* 2009)**.** Aquesta situació es dóna quan hi ha un fragment d'ADN al genoma de referència que no es troba a noves seqüenciacions. Si aquesta regió coincideix amb un element transposable podem assumir que s'ha produït una deleció d'un transposó. Normalment es limita aquesta regió en delecions que ocupin entre uns 200pb i 25 kb per tal de detectar tot tipus de deleció que hagi pogut succeir relacionada amb els transposons.

  Encara que ens referim a aquesta situació com una deleció no necessàriament vol dir que s'hagi delecionat aquesta seqüència sinó que no està present en la mostra reseqüenciada però sí en el genoma de referència. Això pot ser degut per exemple a una inserció al genoma de referència i que en la reseqüenciació estudiada no s'hagi produït una deleció en el genoma reseqüenciat.

- Per altra part trobem les **insercions** (Ewing, 2015)**.** Es donen quan hi ha una determinada seqüència que apareix de nou en noves seqüenciacions però no apareix en el genoma de referència. De nou,

pot ser deguda tant a una inserció en el genoma reseqüenciat com a una deleció en el genoma de referència. Es poden detectar gràcies a la presència de *discordand reads* que mapegen en altres punts del genoma, a més a més aquests *discordant reads* alineen contra un genoma de referència.

Aquestes dues situacions queden descrites en la següent figura:

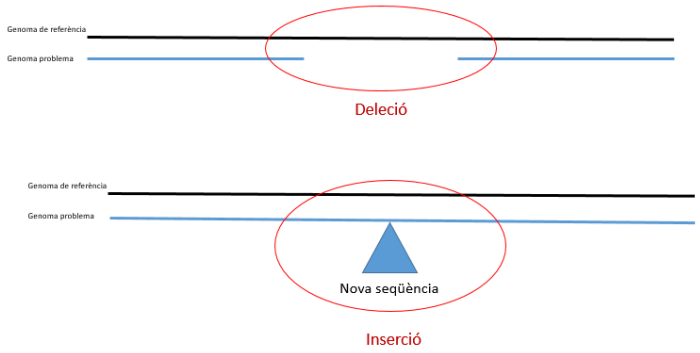

Figura 5: Representació de les delecions i insercions en les noves seqüenciacions

Per tal de poder arribar a detectar aquests polimorfismes, disposant de dades provinents directament de les reseqüenciacions (*raw data)*, és necessari primerament filtrar aquestes dades per qualitat i descartar tots aquells *reads* que no superin els filtres de qualitat. Cal recordar que totes les seqüenciacions a més a més de la seqüència nucleotídica predita tenen associat a cada posició un valor de qualitat. Si aquesta seqüència és de mala qualitat pot alterar els resultats de l'anàlisi. Així mateix per realitzar la seqüenciació són necessaries tot un seguit de seqüències motlle que són necessàries eliminar de l'anàlisi perquè no puguin interferir en els resultats.

Posteriorment serà necessari alinear els *read*s contra el genoma de referència. Hi ha una gran quantitat de programes desenvolupats per dur a terme aquesta tasca (Li and Durbin, 2010). Uns dels més utilitzats són els disponibles en el paquet *BWA* (Li and Durbin, 2010) concretament *BWA Aln i BWA mem*. Aquests permeten alinear correctament contra el genoma de referència les seqüències en format *fastq,* filtrant totes aquelles seqüències que no aliniin en cap regió així com tallant les seqüències en cas de que aliniin parcialment (això serà imprescindible per predir el punt d'inserció d'un transposó polimòrfic). Formant com a *output* un arxiu de format *sam* fàcilment comprimible a un *bam file*. A partir d'aquest arxiu es podrà detectar la presència de polimorfismes causats per transposons sempre i quan es disposin de *discordand reads* en aquest fitxer.

S'observa que hi ha una gran varietat de programes que es poden utilitzar per realitzar tot aquest procés (Ewing, 2015). En aquest treball s'han provat varis programes per realitzar cadascun dels passos tant pel filtratge, alineament i per la detecció de transposons polimòrfics, les proves descrites queden descrites al llarg d'aquest capítol. Els possibles processos seguits per dur a terme la detecció de polimorfismes queden descrits en el següent *workflow:*

Figura 6: *workflow* amb tots els possibles passos per analitzar els polimorfismes de transposons

Per realitzar tot aquest anàlisis s'utilitza la seqüenciació amb nom *V18_DA_B00H6BX.* Aquesta correspon a una mostra de *Physcomitrella patens wt* seqüenciada l'any 2010 amb un *coverage* de 10 vegades la mida del genoma. S'utilitza aquesta mostra ja que, presenta un *coverage* baix i no ocupa molta memòria, això fa que l'arxiu sigui molt més versàtil a l'hora de realitzar tests.

Finalment, tot l'estudi es realitza sota el sistema operatiu Ubuntu Linux 16.04 amb 4gb de memòria RAM un processador Intel Core i5-2500 amb 4 CPU a 3.30 GHz. Per aquells processos més costosos computacionalment es van realitzar en el clúster del servei de bioinformàtica del Centre de Recerca en Agrigenòmica (http://www.cragenomica.es/ca/serveis-cientifics/bioinformatics-core-unit [23/05/2017]), assignant a totes les tasques la mateixa quantitat de memòria per tal d'obtenir resultats comparables, aquells processos que s'han realitzat mitjançant aquest *clúster* es troben descrits en la memòria.

Tot aquest procés va ser utilitzat per estudiar l'evolució dels transposons en l'organisme *Physcomitrella patens*.

L'objectiu d'aquest treball doncs és triple:

Per un costat comprovar quin és el millor mètode per estudiar els polimorfismes de transposició.

Per altra costat, a partir de les seqüenciacions realitzades durant els anys 2011 i 2016 estudiar si hi ha hagut una evolució en quan als polimorfismes de transposons.

Finalment s'estudia l'evolució de *Physcomitrella patens* en varis ecotips.

2.2.- Selecció del mètode de filtratge de les dades (*trimming)* per qualitat

Partint de les dades obtingudes de les seqüenciacions realitzades sobre els clons en els períodes 2010 a 2016 es va seleccionar una d'aquestes mostres, concretament, un clon del 2010 anomenat B00H6BX.

Es disposen de varis arxius corresponents a aquesta reseqüenciació:

- 2 *fastq* que són l'output directe de l'aparell de seqüenciació *illumina* en format *paired-end.*
- També de dades filtrades per qualitat realitzat pel grup de *DNA repair and genome engineering* (el qual va realitzar la seqüenciació).

Es comprova tres processos de filtratge; per una banda, el filtratge prèvi ja realitzat en anterioritat a aquest estudi. Per altra banda, es realitza el filtratge amb el programa *Skewer* (Jiang, *et al.*, 2014) o bé mitjançant *Trimmomatic* (Bolger *et al.*, 2014). Tots els resultats s'han analitzat mitjançant l'aplicació *fastq*c (Andrews, 2010).

*Fastq*c permet realitzar fàcilment una anàlisis de la qualitat de les dades. Obtenint informació de les estadístiques bàsiques; com poden ésser la qualitat per base de les seqüenciacions, el contingut en GC, entre altres. A més indicant quins valors es consideren acceptables quins no i quins presenten certs problemes en quan a la seva qualitat.

Es va procedir a provar la qualitat primerament de les *raw data*, obtenint el següent resultats: observant la qualitat per bases visualitzem una pèrdua de qualitat important al voltant de les posicions 82-86 dels reads.

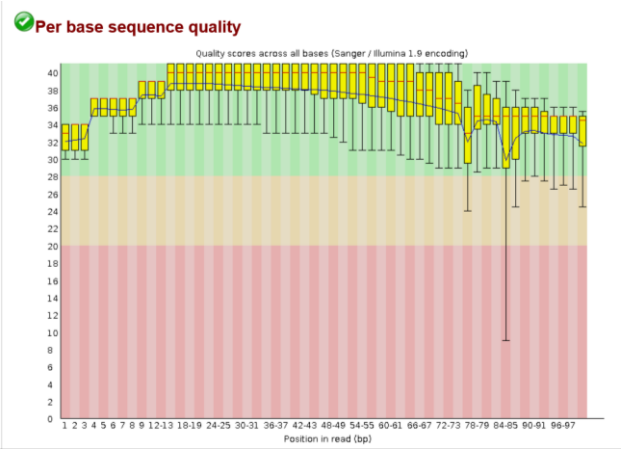

Figura 7: Output de *Fastq*c per la seqüència corresponent a B00H6BX

Observant l'output de *Fastq*c es visualitza que hi ha una pèrdua de qualitat important al voltant de les posicions 84-85. Això s'observa millor amb el gràfic generat *per tile sequence quality:*

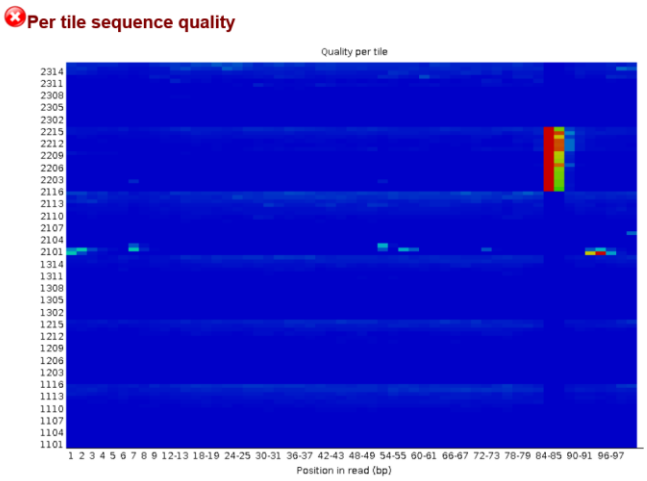

Figura 8: Output de *Fastqc* per tile quality

En aquest gràfic s'observa la qualitat per posicions de les seqüenciacions, en colors blau és que la qualitat és bona, colors més clars significa que hi ha hagut una pèrdua de qualitat, així doncs colors de verd a vermell són indicadors que la qualitat no és bona en aquesta regió. Es visualitza que hi ha tota una regió que té una qualitat molt pèssima, això és per problemes relacionats amb l'aparició de bombolles (https://www.bioinformatics.babraham.ac.uk/projects/fastqc/Help/3%20Analysis%20Modules/12%20Per%20Tile%20Sequence%20Quality.html [03/4/2017]) en el seqüenciador que fan que en determinades zones hi hagi una caiguda de la qualitat. En aquests casos cal filtrar aquesta regió, tallant els reads en aquesta posició.

Mitjançant el procés de filtratge s'intentarà eliminar aquests *reads*, els resultats obtinguts són els següents:

En primer lloc, s'analitzen les dades filtrades prèviament:

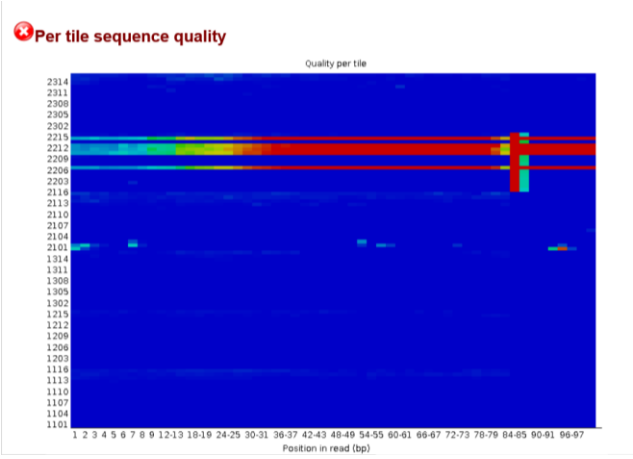

Figura 9: Filtratge realitzat pel grup que va seqüenciar les dades

Veiem que en aquest cas no es resol el problema ja que es va intentar eliminar tots els *read*s que contenien aquest problema en les posicions 84-86

empitjorant molt la qualitat, no solament de la regió, sinó també de tot el procés.

Els resultats del filtratge mitjançant *skewer* són els següents:

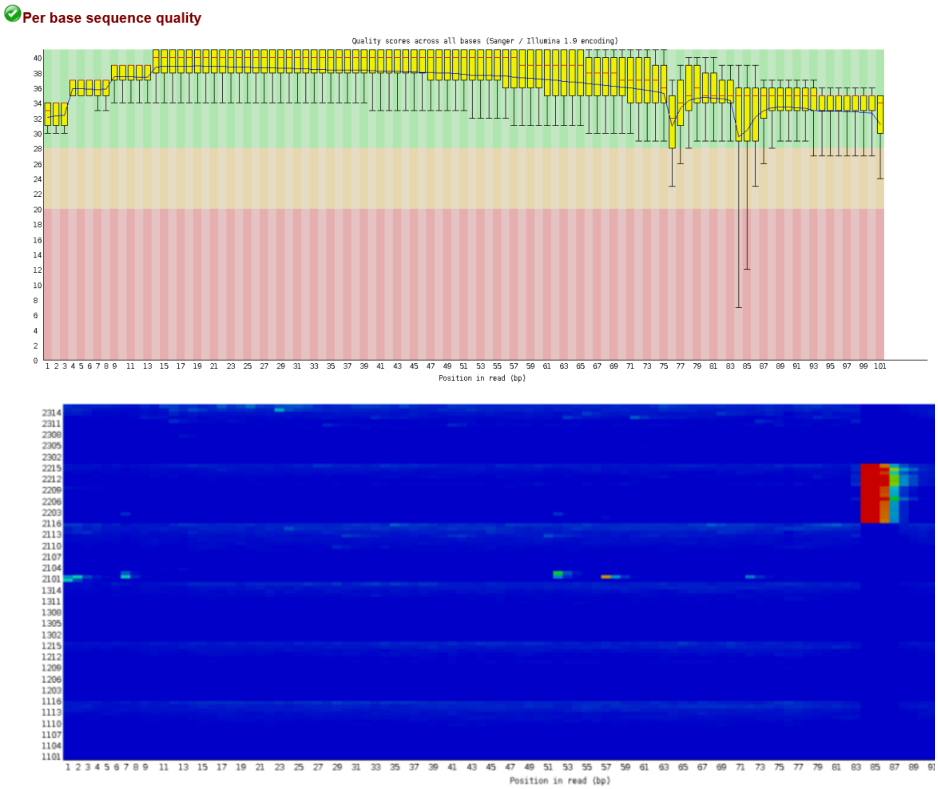

Figura 10: Filtratge amb *skewer* de les seqüències

Veiem que mitjançant *skewer* no es resol el problema obtingut a les posicions 84-86, no solament això sinó que a més presenta nous problemes derivats del filtratge, obtenint nous *warnings* que no teniem abans.

En tercer lloc, es va filtrar mitjançant *Trimmomatic*, amb els següents resultats:

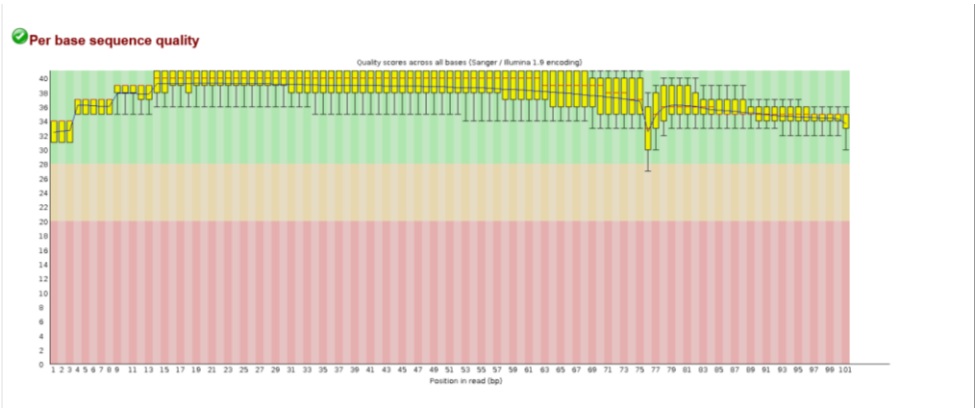

Figura 11: Output de *Fastqc* per el filtratge de *Trimmomatic* sobre les seqüències

Veiem que en aquest cas sí que s'han aconseguit filtrar les seqüències que presentaven una qualitat molt baixa. *Trimmomatic* ha tallat 20 pb en 3' d'aquelles seqüències en què hi havia una qualitat baixa, és a dir aquelles

seqüències que presentaven problemes de qualitat al voltant de les posicions 84-86. Això ho podem visualitzar amb el gràfic *per tile sequence quality:*

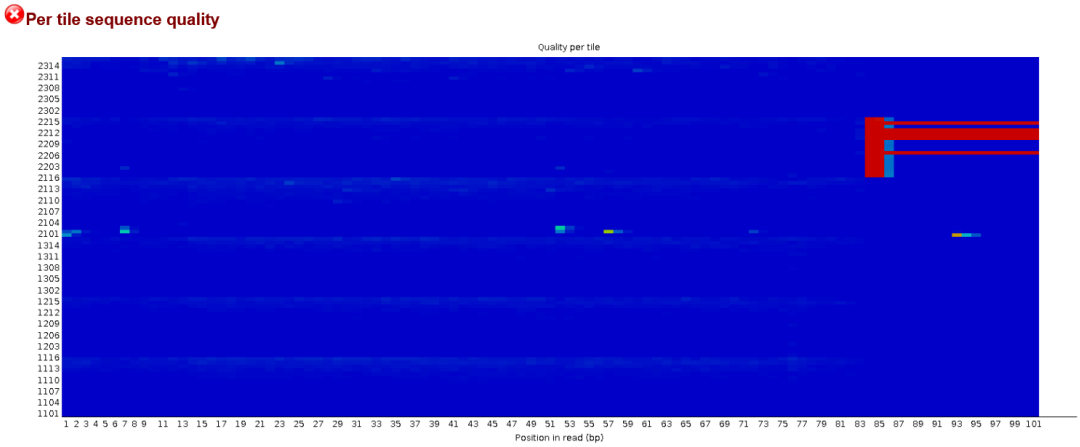

Figura 12: Per tile sequence quality graphic de *trimmomatic*. S'aprecia com s'ha tallat en molts d'aquests casos que la qualitat era pèssima per la posició 84-86.

Això ha permès millorar els resultats obtinguts prèviament, eliminant també els adaptadors necessaris per dur a terme la seqüenciació.

Els resultats obtinguts d'aquesta comparació utilitzant els dos filtratges són els següents:

Taula 1: Comparació dels filtratges de *Skewer* i *Trimmomatic*

Programa	Temps computacional	Reads recuperats
Skewer	00:27:33	99.19%
Trimmomatic	00:32:52	92.36%

Veiem que *skewer* filtra moltes menys seqüències però no millora la qualitat de les dades incials mentre que *trimmomatic* síque hi ha una millora dels resultats tot i filtrar més exhaustivament.

El mètode per filtrar que es selecciona és ***trimmomatic*.**

2.3.-Alineament de les seqüenciacions contra el genoma: *BWA Aln* vs *BWA mem*

Per tal d'alinear les seqüències contra el genoma es presenten dues alternatives: Per una banda utilitzar *BWA Aln* i per altra banda *BWA mem*. Ambdós programes es van comparar per tal de decidir quin era el programa que permet recuperar més *reads* alineats correctament contra el genoma, mantenint la informació dels *discordand reads* imprescindibles per a les anàlisis dels polimorfismes de transposons.

Sota aquests requisits s'elaboren dos *scripts* (disponibles en els annexos III i IV) per tal de córrer *BWA Aln* i *BWA mem* amb les dades obtingudes prèviament i utilitzant el servidor, ja que aquest procés és computacionalment molt més extensiu.

El funcionament de *BWA Aln* (Li and Durbin, 2010) és d'entrada més complexe ja que requereix indexar els arxius mitjançant *BWA index*, posteriorment alinear contra el genoma de referència i finalment generar l'arxiu *sam* mitjançant *sampe,* finalment aquest arxiu es comprimirà en format *bam*. També per la detecció de polimorfismes de transposons és molt important que aquests *read*s estiguin ordenats per posició així que al finalitzar aquests es procediran a ordenar-se mitjançant l'opció *samtools sort* i finalment s'indexen amb *samtools index:*

Per altra banda, l'algorisme de *BWA mem* és més recent (Li, 2013). Es caracteritza en general per la seva capacitat de mapejar una quantitat molt més elevada de *read*s en un temps computacional molt inferior (Li, 2013). Es va procedir a comprovar si això és així per les nostres dates i com funciona per la detecció de polimorfismes de transposició. El funcionament de *BWA Mem* és més senzill ja que realitza automàticament els passos d'indexació, alineament i generació de l'arxiu *sam.* Solament restaria ordenar l'arxiu sam per posició, indexar-lo i comprimir-lo.

Les estadístiques obtingudes per ambdós processos, pel que fa a temps computacional, són les següents:

Taula 2: Dades computacionals de *BWA Aln* i *BWA mem* per realitzar el procés d'alineament

Job	ReqMem	Elapsed	NCPUS	CPUTimeRaw
BWA Aln	4Gc	02:23:52	6	51792
BWA Mem	4Gc	02:27:43	6	53178

A la taula s'indica el *script* seleccionat, la memòria que ha requerit aquest *script* el temps que ha tardat en córrer el programa, el nombre de *CPUs* utilitzat així com la quantitat de memòria utilitzada per cada procés.

S'observa que el programa *BWA Aln* tarda dues hores i 24 minuts aproximat consumint 51792 de temps computacional de la *CPU* mentre que el programa *BWA Mem* ha tardat 2 hores i 28 minuts aproximadament amb una memòria consumida de 53178. Ambdós programes han tardat i han consumit pràcticament el mateix, no observant diferències significatives entre aquests.

Per altra banda, si comprovem els dos processos pel que fa a quantitat de dades recuperades obtenim les següents estadístiques:

Taula 3: Resultats de la comparació de *BWA Aln i BWA mem*

PROGRAMA	*% Reads mapejats*	*Quantitat de singletons*	*Quantitat de discordand reads*
BWA Aln	65.36%	555995	626987
BWA Mem	67.51%	1026365	836666

Recuperem un 65.36% dels *reads* alineats correctament utilitzant *BWA Aln* així com sis-cents mil *discordant reads* aproximadament i cinc-cents mil *singletons.* Mentre que utilitzant *BWA Mem* la quantitat ha estat superior, recuperant un 67.51% del total de *reads* i recuperant un milió de *singletons* i més de vuit-cent mil *discordand reads.*
Cal recuperar, no solament el màxim nombre de *reads alineats,* sinó també la màxima quantitat de *discordand reads* com de *singletons.* Aquests darrers *reads* són imprescindibles per la detecció de polimorfismes de transposons.

La decisió en aquest cas no és evident; per una banda *BWA aln* sembla tardar menys i computacionalment ser menys exigent però per l'altra *BWA mem* alinea molts més *reads* així com *singletons i discordand reads.*

En un primer moment es va decidir que per les diferències mostrades el millor era optar per *BWA mem* al recuperar més *read*s però després d'utilitzar-ho amb els programes de deteccions d'insercions es va optar per *BWA Aln*.

Això va ser degut a que, si l'alineament s'ha produït mitjançant *BWA* mem, els programes de detecció d'insercions de transposons són incapaços de funcionar correctament. Després d'observar això es va preguntar a l'Elizabeth Hénaff, c*read*ora del programa de detecció d'insercions *jitterbug* (Hénaff *et al*., 2015) per les causes d'aquest malfuncionament al córrer *Jitterbug* amb dades procedents de *BWA mem*. La causa és que *BWA mem* conté el que s'anomenen *Split reads*: Aquests són *reads que* poden mapejar a diferents posicions del genoma, fet que dificulta la tasca de detecció d'insercions, no permetent detectar una posició exacte on s'hagi produït aquesta inserció.

Per aquest motiu, es va optar per ***BWA Aln*** ja que no presenta aquest problema.

2.4.-Programes de detecció de polimorfismes: Comparació de programes i selecció

Finalment, obtenim un arxiu *bam* que conté tots els *read*s alineats contra el genoma i llest per poder detectar els possibles polimorfismes d'interès. Tal i com s'ha explicat anteriorment, podem considerar dos tipus de situacions: la primera de totes és que hi hagi una deleció és a dir una seqüència que està en la referència però no en la nova seqüenciació i la segona és que hi hagi una inserció, és a dir, una seqüència que no està en la referència i si en la seqüenciació.

Les estratègies emp*rad*es per ambdues situacions són molt diferents, en aquest cas començarem pel cas més senzill que és la detecció de delecions.

Detecció de delecions:

Els programes de deteccions de delecions (Ye *et al.,* 2009) basen els seus algoritmes en les cerques de *discordand reads* en determinades regions del genoma. En aquests la distància entre ambdós *reads* és superior als 300-400 pb que són habituals. Entre aquests dos *reads* s'observa una regió en què no ha alineat cap *read* degut a l'absència d'aquesta regió. Per trobar els polimorfismes de deleció de transposons s'entrecreuaran les dades amb les anotacions de transposons.

Per tal de detectar les delecions s'opta per utilitzar el programa *Pindel* (Ye *et al.,* 2009). *Pindel* permet identificar aquestes regions gràcies als *discordand reads.*

Aquests mapegen a una distància molt superior a la que hauria de mapejar la parella de *reads* (superior als 400-500pb) obtenint una regió en què hi ha una gran quantitat de *discordand reads* i on a l'interior no hi ha cap *read.* La idea bàsica queda descrita en la següent figura:

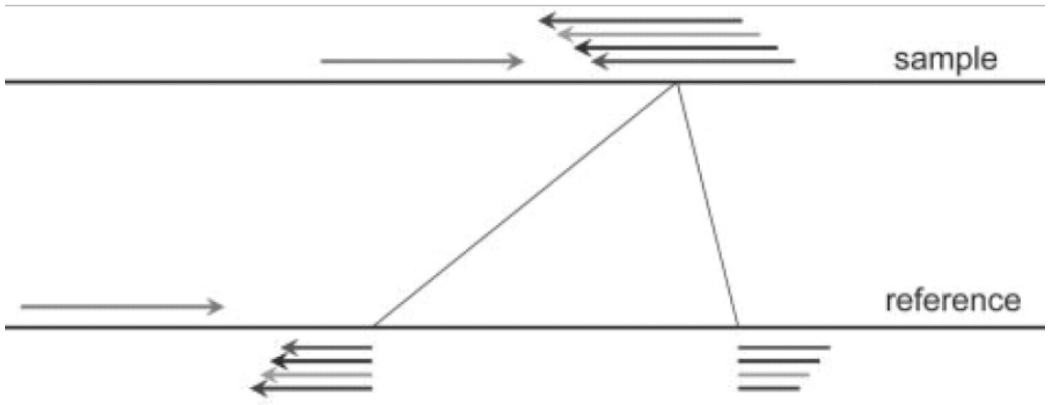

Figura 13: Deleció d'un element detectada per *pindel (Ye et al.,2009)*

Un exemple clar d'una inserció la trobem en la següent visualització mitjançant *IGV* (Thorvaldsdóttir *et al.,* 2013), un visualitzador dels *reads* contra el genoma de referència:

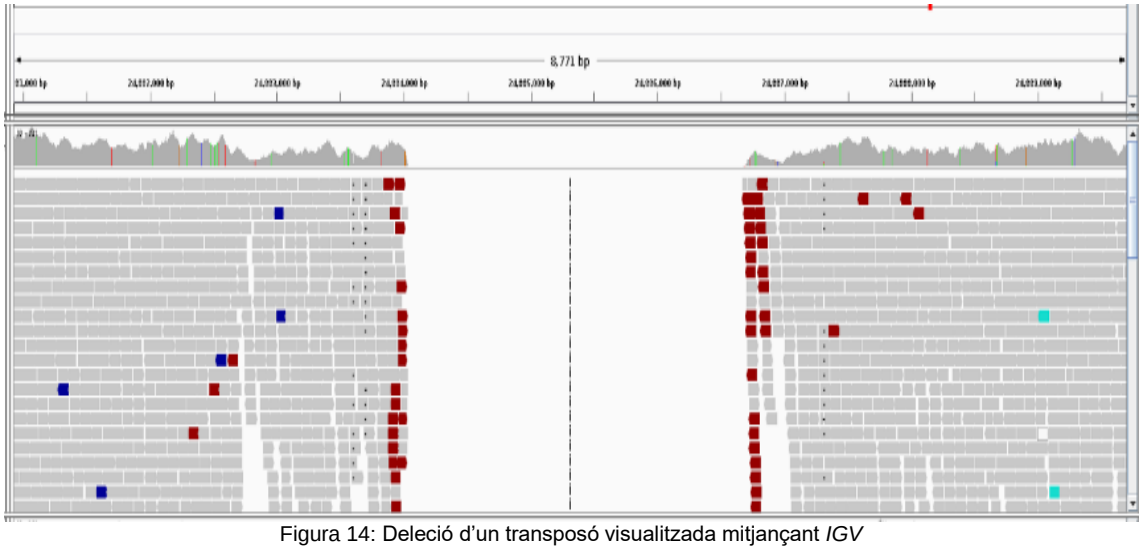

Figura 14: Deleció d'un transposó visualitzada mitjançant *IGV*

Pindel a més a més és capaç de detectar altres canvis estructurals com petites insercions o inversions que en el nostre cas no són d'interès. *Pindel* també és un programa costós computacionalment per això s'executa al servidor mitjançant el *script* descrit en l'annex V.

Mitjançant *Pindel* s'obtenen multiples *outputs* sobre possibles reorganitzacions genòmiques. En aquest cas es varen seleccionar únicament aquelles que coincideixin amb un transposó, per tal de realitzar aquesta tasca, és necessari disposar d'una anotació prèvia de transposons. Aquesta anotació es pot trobar a la pàgina web *Phytozome* *(*https://phytozome.jgi.doe.gov/pz/portal.html#!info?alias=Org_Ppatens *[4/05/2017]).*

Finalment, es va elaborar un *script* per tal d'entrecreuar aquelles delecions que coincideixin amb transposons. En aquest cas ens varem quedar, únicament, amb aquelles delecions més grans de 200 pb i més petites de 25 kb, el *script* es pot trobar en l'annex VI d'aquest treball.

Amb tot aquest procés es va aconseguir obtenir totes aquelles delecions polimòrfiques en les noves reseqüenciacions en format *gff3*.

Detecció de polimorfismes d'insercions:

En aquest cas l'estratègia és completament diferent. Per tal de detectar les possibles insercions hi ha diverses estratègies que utilitzen els programes de detecció de polimorfismes d'insercions (Ewing, 2015) .

Una d'aquestes estratègies utilitzades, partint de dades de *NGS,* es basa en la cerca de *clústers* de zones on hi hagi una gran quantitat de *discordand reads* (és a dir *reads* que mapegin a altres zones del genoma Ewing, 2015)). Un cop identificada aquestes regions es cerca si el *read* que ha estat mapejat en alguna altra seqüència té homologia amb un transposó i amb quin tipus de transposó i finalment prediuen el punt d'inserció. El procés que es realitza és el següent:

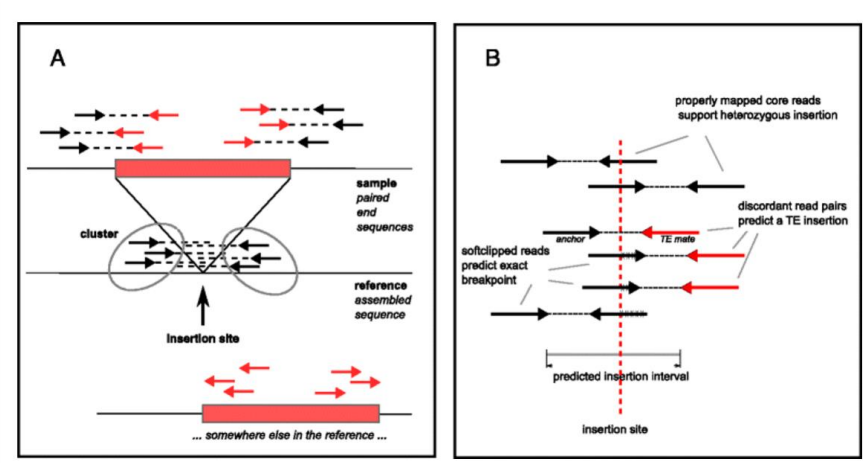

Figura 15: Predicció del punt d'inserció d'un transposó polimòrfic (Hénaff, *et al.*,2015)

Per tal de que aquests programes puguin funcionar correctament és imprescindible que la seqüenciació s'hagi realitzat en format paired-end i que cada *read* es trobi únicament en el genoma com s'ha comentat en l'apartat 1.4, amb això s'aconsegueix definir una zona on es prediu que hi ha una inserció.

Els darrers anys s'han publicat una gran quantitat de programes de detecció de transposons que utilitzen aquest sistema per predir els punts d'inserció, un exemple de la quantitat de programes la trobem a la següent taula publicada(Ewing, 2015):

Taula 4: Programes per la detecció d'insercions d'elements transposables a través de dades de *NGS (Ewing, 2015)*

Name of method	Detection target	Ref.	Notes or use case	Implementation	Availability
TranspoSeq	Transposable elements	[40]	Analysis of Tumour/Normal WGS pairs, extension to analyse WES data as well	Java, R	https://www.broadinstitute.org/cancer/cga/transposeq
Tea	Transposable elements	[65]	Analysis of Tumour/Normal WGS Pairs	R	http://compbio.med.harvard.edu/Tea/
TraFiC	Transposable elements	[66]	Analysis of Tumour/Normal WGS Pairs, detection of transduced sequences	Perl	https://github.com/cancerit/TraFiC
RetroSeq	Transposable elements	[50, 51]	Used for analysis of mouse strain genomes, also demonstrated on human, has genotyping and discovery modes	Perl	https://github.com/tk2/RetroSeq
Tangram	Transposable elements	[75]	Demonstrated on 1000 Genomes Project samples, includes genotyping capability	C, C++	https://github.com/jiantao/Tangram
VariationHunter	Structural Variants	[76, 77]	Among the first methods to detect polymorphic Alu insertions from WGS	C++	http://compbio.cs.sfu.ca/software-variation-hunter
GRIPper	Retrotransposed mRNAs	[78]	Used to detect non-reference gene retrocopy insertions. Demonstrated in humans, mice, and chimpanzees.	Python	https://github.com/adamewing/GRIPper
T-lex/T-lex2	Transposable elements	[52, 53]	Detects both insertions versus the reference and absences of reference elements in other genomes. Demonstrated on Drosophila TEs.	Perl	http://petrov.stanford.edu/cgi-bin/Tlex.html
HYDRA-SV	Structural rearrangements	[79]	General-purpose SV detection, also detects TE insertions	C++, Python	https://github.com/arq5x/Hydra
RelocaTE	Transposable elements	[80]	Demonstrated on mPing insertions in Oryza sativa (rice)	Perl	https://github.com/srobb1/RelocaTE
ITIS	Transposable elements	[81]	Used to detect Tnt1 insertions in Medicago truncatula	Perl	http://bioinformatics.psc.ac.cn/software/ITIS/
ngs_te_mapper	Transposable elements	[82]	Requires TSDs, demonstrated in Drosophila	R	https://github.com/bergmanlab/ngs_te_mapper
TE-Locate	Transposable elements	[83]	Used to examine TE insertions in Arabidopsis populations	Java, Perl	http://sourceforge.net/projects/te-locate/
TIGRA	Structural rearrangements	[84]	Assembly-based SV detection method, demonstrated to identify TE breakpoints	C++	https://bitbucket.org/xianfan/tigra
Mobster	Transposable elements	[85]	Demonstrated on WGS and WES data, Illumina and ABI SOLiD data.	Java	http://sourceforge.net/projects/mobster/
TEMP	Transposable elements	[86]	Geared towards population-level TE detection from pooled data	Perl	https://github.com/JialiUMassWengLab/TEMP
TE-Tracker	Transposable elements	[87]	Attempts to determine source elements in reference. Demonstrated on Arabidopsis.	Perl	http://www.genoscope.cns.fr/externe/tetracker/
Jitterbug	Transposable elements	[41]	Demonstrated on Human and Arabidopsis.	Python	http://sourceforge.net/projects/jitterbug/
DD_DETECTION	Transposable elements	[88]	Does not require input of canonical TE sequences (Database-free)	C++	https://bitbucket.org/mkroon/dd_detection
MELT	Transposable elements	[89]	Used for comprehensive analysis of 2504 participants in the 1000 Genomes Project	Java	http://melt.igs.umaryland.edu/

Entre aquests programes es van seleccionar 4 d'aquests: Primerament *jitterbug* (Hénaff *et al*., 2015) per coneixement previs del seu funcionament en plantes i en el propi genoma de *Physcomitrella patens*, Es va seleccionar també Mobster (Thung *et al.,* 2014), RelocaTE2 (Chen *et al*., 2017) i TE-tracker (Gilly *et al.,* 2014).

Els criteris de selecció per la resta de programes van ser, bàsicament, si s'havien utilitzat prèviament en altres organismes vegetal (Hénaff *et al*., 2015) i si s'havien realitzat prèviament estudis comparatius del seu funcionament (Rishishwar *et al.,* 2016). I finalment, que els programes estesin en revisió constant.

Els programes seleccionats van ser els següents:

Taula 5: Programes seleccionats per establir la comparació

Programa	Utilitzat en plantes	Revisió constant	Estudi comparatiu extern amb altres programes	Any de publicació
Jitterbug	Sí (*A. thaliana)*	Sí	No	2015
RelocaTE2	Sí (*O. sativa)*	Sí	No	2017
Mobster	No	Sí	Sí	2014
TE tracker	Sí (*A. thaliana)*	Sí	No	2014
Retroseq	Sí (*A. thaliana)*	Sí	Sí	2014

El primer programa que es va intentar utilitzar va ser Mobster ja que es disposava d'un estudi previ (Rishishwar *et al.,* 2016) en què es demostrava que era molt efectiu detectant polimorfismes en genomes reals en diferents *coverages.* Sent capaç de detectar amb un nombre relativament baix de falsos positius i falsos negatius un alt nombre de polimorfismes en el genoma humà (Rishishwar *et al.,* 2016).

Es va provar de utilitzar Mobster per detectar les transposicions en *Physcomitrella patens* però Mobster va ser dissenyat per tal de detectar transposicions polimòrfiques en poblacions i no en individus aïllats sobretot aquells MEI (*Mobile Elements Insertion)* relacionats amb malalties humanes, concretament amb el projecte 1000 genomes que consisteix en seqüenciar 1000 genomes diferents humans diferents (Thung *et al.,* 2014). No va ser possible utilitzar-lo amb les nostres dades.

Amb RelocaTE2 succeeix exactament el mateix. Està dissenyat per la detecció de transposicions polimòrfiques de transposons en poblacions. Aquest últim va ser dissenyat per comprovar polimorfismes en plantes concretament en Oryza sativa (arròs) (Chen *et al*., 2017), tot i això no ens és útil tampoc per les nostres dades.
Es va comprovar que una gran quantitat d'aquests programes estan pensats per tal de trobar poblacions polimòrfiques de determinats transposons (Rishishwar *et al.,* 2016) no sent útils pel nostre estudi.

Es va optar per comparar amb als altres dos programes *TE tracker* (Gilly *et al.,* 2014). i Retroseq (Gilly *et al.*, 2014) i comparar-los amb la detecció de les dades amb *Jitterbug* (Hénaff *et al*., 2015).

Els tres programes tenen un funcionament molt similar sent capaços de detectar insercions de transposons a través de les seqüenciacions en formats paired-end en clons individuals.

Pel que fa al rendiment dels tres programes , s'obtenen resultats molt diferents amb la capacitat de predir insercions, *TE-Tracker* comparat amb *retroseq* amb el genoma de *A. thaliana* s'obtenen els següents resultats :

Taula 6: Rendiment de TE- tracker comparat amb retroseq (Gilly *et al*., 2014).

Software	# Insertion + donor found	# Insertion + normal donor found	# Insertion + composite donor found	# Insertion + long donor found	# Insertion + short donor found
RetroSeq	128 (43%)	87 (87%)	0 (0%)	0 (0%)	41 (82%)
TE-Tracker	257 (86%)	91 (91%)	81 (81%)	42 (84%)	43 (86%)

Jitterbug també es va comprovar el rendiment en *A. thaliana* comparant-ho amb *Retroseq*, obtenint els resultats següents:

Taula 7: Rendiment de *Jitterbug* comparat amb retroseq (Hénaff *et al*., 2015).

		PPV (%)	Sensitivity (%)
Jitterbug	raw	37.16	89.72
	filtered	92.7	85.05
RetroSeq	extended +/- 100 bp and merged	61.01	90.26
	extended, merged and filtered	87.31	88.21

No va ser possible realitzar la anàlisis amb *Retroseq i TE tracker*. Això és degut a que ambdós programes per realitzar la detecció d'insercions l'output del programa *Repeatmasker* (Tarailo-Graovac *et al*., 2009). *Repeatmasker* genera com a *output* una taula amb tots els elements transposables detectats en aquest període de temps, és un format específic d'aquest programa.

En el cas de l'anotació de transposons de la qual es disposa per fer aquest estudi en *Physcomitrella patens* es va generar mitjançant la *pipeline Repeatmasker* combinada amb altres programes com *LTR harvest* (Ellinghaus *et al.*, 2008), entre altres. Aquest és un fitxer en format *gff3* que conté una anotació d'elements transposables molt més complerta que la generada per *Repeatmasker* (Rensing *et al*., 2017).

Per tal de poder comparar aquests programes, es va intentar adaptar les dades disponibles a format *Repeatmasker* però no es va trobar cap programa que permetés fer la conversió de *gff3 a l'output de Repeatmasker*. Això era causat, per les complexitats de les taules contenint valors qualitatius. Per aquest fet no ha estat possible realitzar aquesta comparació.

Jitterbug en canvi sí que permet utilitzar com a *input* arxius en format *gff3* per realitzar la detecció de polimorfismes de transposons, per aquest fet es va utilitzar aquest programa per detectar els polimorfismes d'insercions.

Les comandes utilitzades per l'execució del programa són les següents:

```
$jitterbug.py --psorted V18_DA_B00H6BX.bam -t Ppatens_v3.0_251_TEanno_v3__5Oct2015.gff3 -l
B00H6BX -n Name -o B00H6BX/V18_DA_B00H6BX.TE_insertions_paired_clusters.gff3

$jitterbug_filter_results_func.py -g B00H6BX/V18_DA_B00H6BX.TE_insertions_paired_clusters.gff3 -
c B00H6BX/V18_DA_B00H6BX.filter_config.txt -o
B00H6BX/V18_DA_B00H6BX.TE_insertions_paired_clusters_filtered.gff3
```

Figura 16: Comandes utilitzades per executar *jitterbug* sobre les dades d'interès

Finalment, es va obtenir un *output* amb els transposons detectats. Els resultats d'aquests estan descrits en l'annex VII conjuntament amb la resta de clons individuals durant el període 2008 al 2016. Bàsicament, es detecten 8 transposons però un cop s'aplica el filtre de *jitterbug* no queda cap d'aquests transposons possiblement polimòrfics.

Jitterbug també proporciona dos outputs interessants; per una banda, les estadístiques dels *read*s i ,per altra banda, estadístiques sobre el temps computacional del programa. Les dades obtingudes són les següents:

Taula 8: Estadístiques obtingudes per *Jitterbug*

Mostra	*Coverage*	Temps	Nº Cpu	Mitjana de la longitud dels fragments	SD de la longitud dels fragments	Longitud dels *read*s	SD dels *read*s
V18_DA_B00H6BX	13.0172	0:02:09.66	1	373.51 pb	83.83	95.59	12.89

A la taula observem el *coverage* de la mostra que és de 13 vegades el genoma de *P. patens*, va tardar 2 minuts i 9 segons en realitzar la detecció destinant

únicament un *CPU* per realitzar la detecció. La mitjana de la distància entre *reads és de 373 pb* amb una desviació de 83.83 pb. Mentre que la longitud dels *reads* és de 95 pb de mitjana amb una desviació estàndard de 13 pb.

Finalment, el programa seleccionat per detectar polimorfismes de transposons d'inserció va ser ***Jitterbug.***

2.5.-Metodologia escollida

Amb tota la informació obtinguda en els anteriors apartats es defineix un *Workflow* per tal detectar el màxim nombre possible de polimorfismes causats per transposons en un individu. S'ha realitzat tot el procés per intentar maximitzar per una banda el màxim de *reads* de qualitat correctament alineats així com el màxim nombre possible de *Discordand reads* imprescindibles, com hem pogut veure, per tal de trobar els polimorfismes d'inserció de transposons.

És imprescindible definir prèviament el problema a identificar ja que com hem pogut veure hi ha multitud de programes de detecció de polimorfismes de transposons cadascú amb una utilitat diferent. Depenent del problema plantejat aquesta metodologia pot variar, però com hem pogut veure per tal d'identificar polimorfismes en individu aquesta és la més idònia.
El *workflow* escollit és el següent:

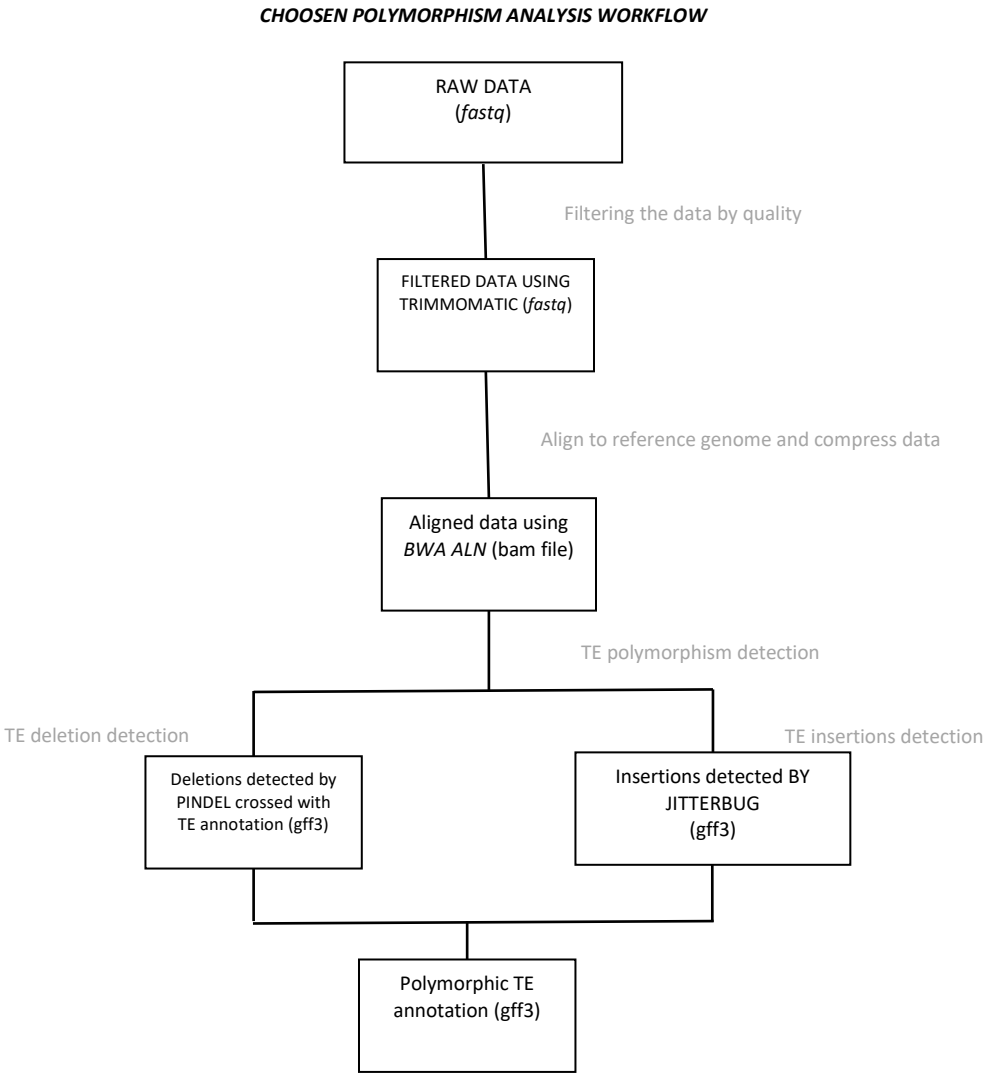

Figura 17: Workflow utiltizat al llarg de tot el treball per la detecció de polimorfismes

3.- Detecció de polimorfismes del genoma de *P. patens* en diferents ecotips

Gràcies a tot el procés desenvolupat en l'anterior capítol d'anàlisis i selecció de programes, s'ha aconseguit descriure un *workflow* que permet la detecció de polimorfismes d'elements transposables.

Es disposa de les dades de dos ecotips de *P. patens*:

- Ecotip *Villersexel:* Aquest ecotip ja havia estat seqüenciat i s'havien descrit els polimorfismes de transposons amb anterioritat (Rensing *et al*., 2017). Es pretén però, detectar els polimorfismes de transposons dintre de transposons els quals no s'havien detectat fins el moment.

- Ecotip *Reute:* Les dades de la seqüenciació estan disponibles a http://www.ebi.ac.uk/ o *European Nucleotide Archive* en format *fastq*. S'han estudiat els polimorfismes causats per *SNPs* però no els polimorfismes de transposons (Hiss *et al*., 2017).

Amb les dades obtingudes es pot realitzar una anàlisis de l'evolució de *Physcomitrella patens* amb els tres ecotips: *Gransden* que és el genoma de referència, *Villersexel* i finalment *Reute*.

L'ecotip *Villersexel* va ser descrit a França (Rensing *et al*., 2017) mentre que l'ecotip *Reute* va ser estudiat a Alemanya (Hiss *et al*., 2017). Hi ha una gran quantitat d'ecotips de *Physcomitrella patens* que s'estan estudiant però a dia d'avui solament hi ha disponibles les dades, de forma pública, d'aquests dos ecotips a la pàgina web *European Nucleotide Archive*.

A més a més, es disposen de dades on es comparen la variació nucleotídica entre els tres ecotips. Aquests van ser descrits fa poc (Hiss *et al.*, *2016).* En aquest estudi no es van analitzar les diferències possibles en els elements transposables polimòrfics entre els tres ecotips.

3.1.- Estratègia seguida per realitzar l'anotació

Per tal de generar l'anotació en *Reute* i detectar nous polimorfismes d'inserció en *Villersexel* dintre de transposons es va realitzar el següent procés:

- S'han generat els *bam files* seguint el *workflow* descrit en l'apartat 1, en cas que no es disposés d'aquests és a dir:
    - S'ha filtrat utilitzant *trimmomatic*
    - S'ha alineat contra el genoma de referència utilitzant *BWA aln*
- S'ha comprovat el *coverage* de les mostres que sigui entre 20 i 40 vegades la mida del genoma
- S'ha llençat *Jitterbug* i *pindel* per tal de detectar els polimorfismes dels transposons
- S'ha llençat *jitterbug* amb una llibreria parcial contenint únicament els elements *RLG1*
- S'ha llençat *jitterbug* amb una llibreria parcial de transposons contenint tots els elements transposables amb excepció dels elements *RLG1*

Amb tot s'ha generat l'anotació dels polimorfismes de transposons en *Villersexel* i *Reute*, el procés i els resultats està descrit en els següents apartats.

3.2.- Detecció de polimorfismes d'inserció dintre de transposons

Cal recordar inicialment la definició d'insercions i delecions que hem descrit en el primer capítol una **inserció** s'entén com una nova seqüència en un dels ecotips que no està en el genoma de referència i una **deleció** és una seqüència que està en el genoma de referència i no està en els diferents ecotips.

El genoma de *Physcomitrella patens*, com s'ha comentat a l'inici d'aquest treball és altament repetitiu, tenint un 25% del genoma ocupat per elements transposables i d'aquests un 90% són elements *RLG1*.

Els elements *RLG1* són d'especial interès en *P. patens*, aquests són elements de tipus *Gypsy* i presenten un *Chromodomain* que els dirigeix cap a zones heterocromàtiques. La seva amplificació al llarg de tot el genoma en aquestes zones els fa d'especial interès per estudiar l'impacte dels transposons en la formació d'heterocromatina (Rensing *et al.,* 2017).

Jitterbug (Hénaff *et al*., 2015), el programa que s'utilitza en aquest treball per la detecció dels polimorfismes d'inserció, no detecta els polimorfismes a l'interior d'altres transposons. Això és degut a que ambdós *discordant reads* mapejen en transposons, i per tant en zones repetitives, podent-se situar en múltiples punts del genoma. Això incrementa molt la possibilitat de que el que estem detectant siguin realment artefactes enlloc d'insercions reals.

Que els elements *RLG1* tendeixin a inserir-se a aquestes regions suposa un repte per la seva detecció, ja que *Jitterbug* (Hénaff, Elizabeth, *et al*., 2015) únicament pot detectar els polimorfismes de transposons en zones úniques.

També la informació d'aquest article (Rensing *et al*., 2017), apunta a que els elements *RLG1* no són únicament els més abundants sinó també els elements transposables més expressats en el genoma de *P. patens.*

Jitterbug, tal i com s'ha comentat, no està dissenyat per detectar insercions dintre d'altres elements transposables. Cal pensar una estratègia per forçar aquest programa a detectar insercions dintre d'elements transposables.

L'estratègia que es va realitzar va ser executar *Jitterbug* amb llibreries parcials de transposons, és a dir llibreries que no contenen totes les anotacions sinó que s'han exclòs determinats elements (com per exemple els elements *RLG1* o la resta d'elements transposables). Això permet detectar polimorfismes de transposons dintre de transposons a costa d'una pèrdua de sensibilitat important, incrementant el risc de detecció d'artefactes.

Per tal d'intentar solucionar aquest problema es va plantejar filtrar amb més rigor. Sabem que *Physcomitrella patens* es comporta com un organisme haploide durant gran part del seu cicle vital i que el teixit seqüenciat és haploide. Això fa que no s'espera trobar transposicions heterozigotes a la població. Per això a l'hora de detectar la zigositat establim que aquesta com a mínim sigui d'un 75%.

Un resum gràfic del que plantegem resoldre és el de la següent figura:

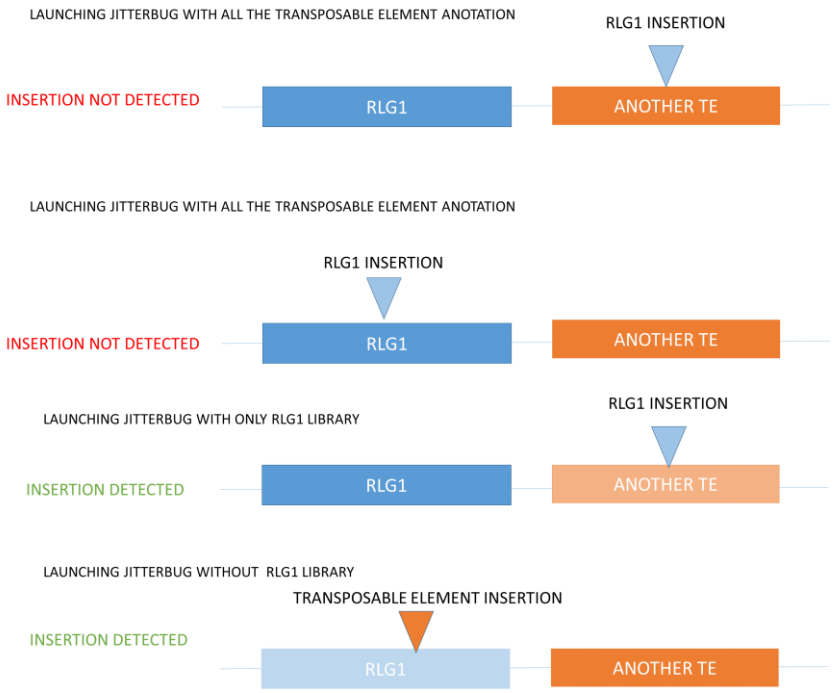

Figura 18: Detecció de transposons dintre d'altres transposons, plantejament seguit

Amb tot el procés realitzat s'espera detectar noves insercions d'elements transposables dintre d'altres elements, i que els que trobem amb més abundància siguin els elements *RLG1.*

Els passos seguits són els següents:

1) Crear una llibreria que contingui únicament els elements *RLG1*
2) Llençar *jitterbug*
3) Anàlisis dels resultats

Les llibreries d'elements parcials es van elaborar seleccionant determinats elements de l'anotació d'elements transposables.
Bàsicament es van elaborar dos llibreries:

- Una primera llibreria que contenia tots els elements transposables amb excepció dels elements *RLG1*. Això permet la detecció d'insercions d'altres elements transposables a l'interior dels elements *RLG1.*
- Una segona llibreria que contenia únicament els elements *RLG1;* permetent la detecció d'insercions d'elements *RLG1* dintre d'altres elements transposables.

Seguidament es va llençar amb *jitterbug* i filtrar els resultats, descrits en l'annex VIII.

Els resultats obtinguts, després d'executar *jitterbug* per l'ecotip *Villersexel i Reute* queden descrits en els capítols 3.3 i 3.5 d'aquesta memòria.

S'esperava detectar una gran quantitat d'elements *RLG1* polimòrfics dintre d'altres elements transposables mentre que no esperàvem detectar una quantitat significativa dels altres elements transposables dintre dels elements *RLG1*, pels estudis que s'han realitzat prèviament (Rensing *et al*., 2017). Efectivament, tal i com veurem en els següents capítols observem que això és així.

Per finalitzar aquest capítol, cal dir que tots els resultats s'han filtrat per una zigositat superior al 75%. Això és degut a que tal com s'ha comentat *P. patens* és un organisme haploide durant la gran majoria del seu cicle vital, per aquest fet no esperem polimorfismes que estiguin en homozigosis. Com que el risc de detectar artefactes en aquest cas és molt més alt, filtrant per una zigositat superior al 75% filtrem tots aquells possibles artefactes que hagin pogut aparèixer en aquest procés.

3.3.-Detecció de transposons polimòrfics en zones repetitives en l'ecotip Villersexel

La detecció de transposons polimòrfics en *Villersexel* va ser realitzada en anterioritat a aquest estudi (Rensing *et al*., 2017). Aquesta està pendent de publicació actualment. L'ecotip *Villersexel* presenta una diversitat elevada amb l'ecotip de referència *Gransden.* Presenta per exemple una gran quantitat de *SNPs* comparat amb el genoma de referència(Hiss *et al.*, *2016):*

Pel que fa els polimorfismes detectats es van obtenir els següents resultats (Rensing *et al*., 2017):

Taula 9: Polimorfismes en l'ecotip *Villersexel*

ECOTIP	Total polimorfismes	Insercions detectades per Jitterbug	Delecions detectades per pindel
Villersexel	1240	298	942

Com observem en la taula anterior, hi ha una gran diferència entre el nombre d'insercions detectades amb el nombre de delecions. Això pot ser degut a que no s'han detectat totes aquelles insercions de transposons dintre d'altres elements repetitius. Per tal de detectar aquestes insercions es va forçar *Jitterbug* a detectar aquests polimorfismes utilitzant l'estratègia del capítol 3.2 d'aquesta memòria.

Finalment, es varen filtrar tots aquells polimorfismes que coincideixin amb els transposons polimòrfics dels períodes 2010 a 2016 (dades generades en el capítol 4 d'aquesta memòria). Aconseguint eliminar així falsos positius causats per errors en la seqüenciació i anotació del genoma.

Els resultats obtinguts d'aquest procés són els següents:

Taula 10: Polimorfismes en l'ecotip Villersexel després de detectar insercions dintre de transposons

ECOTIP	Total polimorfismes	Insercions detectades per Jitterbug	Insercions TE dintre *RLG1*	Insercions *RLG1* dintre altres TE	Delecions detectades per pindel
Villersexel	1854	298	4	594	942

Veiem que hem aconseguit recuperar un total de 594 polimorfismes d'inserció d'elements *RLG1* dintre d'altres transposons. Això reafirma que aquests elements tenen tendència a insertar-se dintre de transposons ja que únicament detectem 4 transposons no *RLG1* que s'han insertat dintre de *RLG1*.

S'ha aconseguit incrementar pràcticament un 300% la quantitat de polimorfismes detectats. El 94% d'aquests nous polimorfismes detectats són *RLG1* polimòrfics a l'interior d'altres transposons i solament un 6% corresponen a altres transposons que cauen dintre d'un element *RLG1*.

En *Physcomitrella patens* es creu que els transposons juguen un paper clau sobre la formació de l'heterocromatina (Rensing *et al*., 2017). Per tal de saber l'impacte d'aquests transposons és de gran rellevància l'estudi dels polimorfismes en *Physcomitrella patens* entre aquests elements trobem els elements *RLG1* que tenen un *chromodomain* relacionat amb les zones heterocromàtiques (Rensing *et al*., 2017). Aquestes dades generades ens poden servir per estudiar l'impacte dels transposons en aquestes zones.

3.4.-Validació de l'anotació en l'ecotip *Villersexel*

Per tal de verificar els resultats es varen dissenyar 8 encebadors flanquejant quatre elements transposables polimòrfics *RLG1* que cauen dintre de transposons (les seqüencies dels quals i les condicions en què amplifiquen es troben en l'annex VIII). Aquests encebadors s'han dissenyat mitjançant el programa *Primer3plus.*
*(*http://www.bioinformatics.nl/cgi-bin/primer3plus/primer3plus.cgi *[21/04/2017]).*

La metodologia seguida per tal de detectar els polimorfismes ha estat en tot cas la mateixa. Primerament s'ha visualitzat els *reads* mitjançant *IGV* (Thorvaldsdóttir *et al.,* 2013) i seleccionat la regió. Un cop definida la regió es varen dissenyar encebadors amb el programa *Primer3plus.* Finalment, com que es tracta en la majoria de casos de regions molt repetitives es realitzar un Blast (Altschul *et al.,* 1990) comprovant que eren seqüències úniques. Un cop realitzades aquestes comprovacions es varen realitzar les *PCR* per verificar els resultats.

Es varen analitzar 4 d'aquests polimorfismes verificant que els 4 eren realment polimòrfics (annex VIII), en aquest cas per sintetitzar solament es descriurà el disseny i verificació d'una d'aquestes verificacions:

Els quatre polimorfismes estaven situats en illes de transposons.
Es van dissenyar *primers* per tal d'amplificar un element *RLG1* polimòrfic en *Villersexel*. Com a exemple, es va seleccionar en aquest cas un polimorfisme en el cromosoma 7 que cau molt proper al gen Pp3c7_6520 pel seu possible impacte sobre el gen . Per tal de verificar si era realment una inserció o no es va realitzar *PCR* sobre mostres tant de Gransden com de *Villersexel* (les condicions d'amplificació queden descrites en l'annex VIII).
Les *PCR* es van fer utilitzant la polimerasa *Long amp* que permet amplificar fragments grans d'*ADN.* Les condicions d'amplificació queden descrites en l'annex VIII.

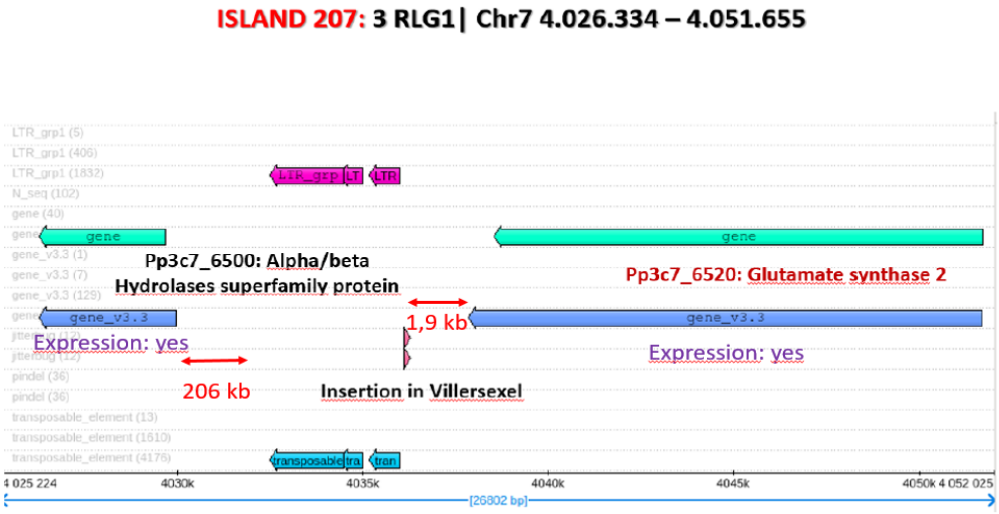

Figura 19: Inserció d'un element *RLG1* proper a un gen, es comprovarà mitjançant *PCR* si és o no polimòrfic

En *Gransden* esperaríem que amplifiqués un fragment de 240 pb ja que el producte predit *in silico* en el genoma de referència (és a dir sense l'inserció) és d'aquesta mida. Mentre que en *Villersexel* esperaríem com a mínim que tingués la mida d'un element *RLG1* és a dir unes 7-8 kb en cas que sigui un element *RLG1* complert.
El resultat obtingut és el següent:

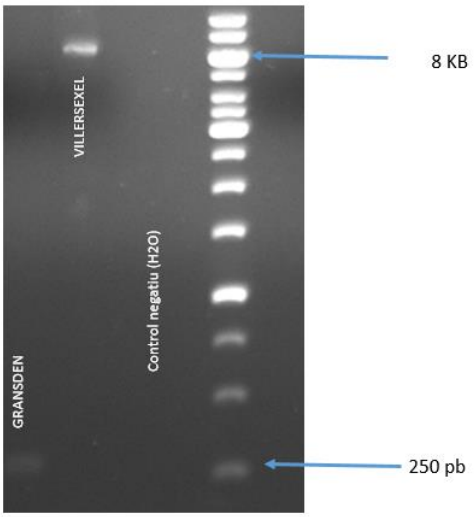

Figura 20: Polimorfisme d'element *RLG1* observem una banda d'uns 250 pb en gransden i d'unes 8 kb en *Villersexel*

Així doncs s'observa una banda de 250 pb en *Gransden* que correspon al fragment d'ADN genòmic sense que s'hagi produït la inserció i en *Villersexel* s'observa una banda d'ADN de 8 kb aproximadament, fet que fa suposar que si que s'ha produït una inserció en aquest punt. Per tal de verificar-ho es va purificar el producte de *PCR* mitjançant el kit *PCR Purification Kit (Qiagen)* i es va seqüenciar utilitzant els oligonucleòtids flanquejants, obtenint una seqüència d'uns 800 pb.

La seqüenciació es va alinear tant contra el genoma com contra els elements *RLG1*. Això es va realitzar mitjançant el programa Blastn del *NCBI* (https://blast.ncbi.nlm.nih.gov/Blast.cgi?PAGE_TYPE=BlastSearch). El *BLAST* (Tatsutova, *et al.,* 1999) de la seqüenciació amb els elements *RLG1* es va obtenir una homologia d'un 98% amb aquests elements, corresponent al *LTR* d'aquest element. Així també presentava una homologia del 96% per altra banda amb la seqüència del genoma on s'ha predit la inserció.

En aquest cas, s'ha pogut visualitzar inclús les *target side duplications* causades per la inserció del transposó.

Aquest exemple, ha permès comprovar que aquests nous polimorfismes d'insercions són realment polimorfismes i no són artefactes que haguem pogut detectar.

3.5.-Anotació de polimorfismes de transposons en l'ecotip *Reute*

En el cas de l'ecotip *Reute* no es disposa de cap dada sobre la detecció de transposons en aquest ecotip. Es va procedir a realitzar aquesta detecció realitzant tot el procés descrit en el capítol anterior.

Es disposa d'una seqüenciació en format *paired- end* pública obtinguda del següent enllaç: http://www.ebi.ac.uk/ena/data/view/SRR3099021&display=html

En aquest cas es disposa d'una seqüenciació en format *Illumina Paired end*. S'utilitza el *workflow* que hem dissenyat en el primer capítol per tal de detectar els transposons en aquesta seqüencia.

Les estadístiques obtingudes de la generació del *bam file* mitjançant *samtools flagstat* són les següents:

```
270527008 + 0 in total (QC-passed reads + QC-failed reads)
0 + 0 secondary
0 + 0 supplementary
0 + 0 duplicates
223780239 + 0 mapped (82.72% : N/A)
270527008 + 0 paired in sequencing
135263504 + 0 read1
135263504 + 0 read2
218625627 + 0 properly paired (80.81% : N/A)
223008453 + 0 with itself and mate mapped
771786 + 0 singletons (0.29% : N/A)
3664270 + 0 with mate mapped to a different chr
```

Figura 21: Output obtingut de *samtools flagstat*

Es recuperen un 83% aproximadament del total de *reads* que han aconseguit *mapejar* de forma correcta contra el genoma de referència. També recuperem un nombre important de *discordand reads* i *singletons,* gairebé 4 milions. Aquests ens permetran predir els polimorfismes de transposons.

El *coverage* en aquest cas és prou elevat sent d'un 54.744 vegades la mida del genoma.

Les estadístiques obtingudes per *jitterbug* dels *reads són les següents:*

```
fragment_length 236.12
fragment_length_SD 63.93
read_length 95.76
read_length_SD 11.09
```

Figura 22: Longitud dels *reads* així com mida dels fragments

Aquestes estadístiques retornen la informació de la longitud mitjana dels fragments en format *paired end* i la desviació estàndard d'aquests fragments. També proporcionen informació sobre la longitud en parell de bases de cadascun dels *reads*; calculant la mitjana d'aquests *reads* i la seva desviació estàndard.

Finalment, s'obtenen els següents resultats pel que fa a polimorfismes en *Reute*:

Taula 11: Polimorfismes en l'ecotip *Reute*

ECOTIP	Total polimorfismes	Insercions detectades per *Jitterbug*	Delecions detectades per *pindel*
Reute	140	38	102

En *Reute* es recuperen un total de 140 polimorfismes de transposons. Aquest valor comparat amb l'ecotip *Villersexel* és baix, ja que en aquest darrer havíem recuperat un total de 2400 polimorfismes de transposons.

En aquest ecotip (*Villersexel*) veiem que el nombre tant d'insercions com de delecions és molt superior. S'arriba a la mateixa conclusió observant la quantitat de *SNP*s en *Villersexel* i *Reute* tal i com mostra la següent taula de l'article de Hiss, Manuel, *et al* (2017)

Taula 12: Estudi de *SNP* i indels en els ecotips *Villersexel* i *Reute* (Hiss *et al* ,2017)

Table 2 Summary table of SNP effect analysis for *Physcomitrella patens* ecotypes Reute and Villersexel, as compared with Gransden

	Reute – SNPs		Reute – Indels		Villersexel – SNPs		Villersexel – Indels	
Type	Count	Percent	Count	Percent	Count	Percent	Count	Percent
Intergenic	249 609	82.6	13 560	59.0	2 355 783	81.1	155 247	63.0
Upstream	18 628	6.17	3729	16.2	200 512	6.91	39 690	16.1
Downstream	17 753	5.88	2835	12.3	196 593	6.77	32 902	13.4
Intron	5132	1.70	1139	4.96	50 725	1.75	7868	3.19
Exon	4542	1.50	381	1.66	38 405	1.32	1673	0.68
5′-UTR	2508	0.83	507	2.21	22 546	0.78	3292	1.34
3′-UTR	2272	0.75	487	2.12	22 994	0.79	3280	1.33
Other	1683	0.56	337	1.47	16 022	0.55	2356	0.96
Sum	302 127		22 975		2 903 580		246 308	

Tal i com observem a la taula s'han detectat un total de 302127 SNPs en *Reute* mentre que en *Villersexel* el nombre és molt superior (un total de 2903580 polimorfismes).

Els resultats són similars els observats en les anàlisis descrites en aquest treball en el qual hem detectat molts més polimorfismes en l'ecotip *Villersexel* que en l'ecotip *Reute.*

Tots els gff3 generats s'adjunten amb el treball i es poden visualitzar a la pàgina web següent:
https://genomevolution.org/coge/GenomeView.pl?gid=33928

Tots aquests arxius de polimorfismes es van filtrar sota les mateixes condicions. És a dir per una zigositat superior al 75% i per tal de verificar que no eren artefactes, es van entrecreuar amb els polimorfismes obtinguts en el següent capítol i es van descartar de la detecció ja que no esperem que hi hagi polimorfismes comuns entre noves seqüenciacions de l'ecotip de referència i els diferents ecotips descrits en aquest capítol.

4.- Anàlisis de l'evolució del genoma de *Physcomitrella patens* durant els darrers 8 anys

4.1.-Introducció

Per tal de comprovar l'activitat *off- target* del sistema *CRISPR-Cas9* (sistema d'edició genòmica dirigit) es van seqüenciar mostres *de Physcomitrella patens* tractades amb el sistema i com a control es van seqüenciar també mostres control sense tractar. Es va observar que les mostres control presentaven diferències de *SNP*s respecte la seqüenciació original del any 2008 tot i provenir del mateix clon propagat vegetativament al laboratori. Vist això es va recultivar teixit del any 2011 de les mateixes mostres, veient que hi havia *SNP*s que estaven presents el 2011 i no el 2008 però sí el 2016 així com mostres presents el 2016 i no presents ni el 2008 ni el 2011.

Aparentment durant els darrers 8 anys i segons les dades obtingudes d'un anàlisis preliminar dels *SNPs* del grup *DNA repair and genome engineering de Versailles-Grignon* sobre aquestes dades, sembla que s'ha produït un procés evolutiu. Aquest estudi preté comprovar si s'han generat nous polimorfismes de transposons, bé insercions d'elements o bé delecions d'alguns elements per recombinació homòloga.

Per tal de verificar aquesta hipòtesis es disposa de 12 seqüenciacions, aquestes queden descrites en la següent taula:

Taula 13: mostres seqüenciades en els diferents períodes de temps

Mostra	Any	Clon	Mostra	Any	Clon
V18_DA_B00H6BX	2010	Wt 1	V18_DA_B00H6C9	2016	Cas9-2
V18_DA_B00H6BY	2010	Wt 1	V18_DA_B00H6CA	2016	Cas9-2
V18_DA_B00H6BZ	2010	Mn 1	V18_DA_B00H6C2	2016	Wt 3
V18_DA_B00H6C0	2010	Mn 1	V18_DA_B00H6C3	2016	Wt 3
V18_DA_B00H6C1	2016	Wt 2	V18_DA_B00H6C4	2016	Cas9-3
V18_DA_B00H6C8	2016	Wt 2	V18_DA_B00H6C6	2016	Cas9-3

Com veiem es disposa de 4 seqüenciacions del any 2010 provinents d'un únic clon; 2 pertanyents a mostres *wild type* 2 a tractades amb meganucleases (també per comprovar l'activitat offtarget). Es disposen de 8 seqüenciacions del any 2016 provinents de 2 clons. Cadascun d'aquests es va tractar amb CRISPR-Cas9 i es va seqüenciar la mostra tractada i sense tractar.

En aquest cas , es compara la comparació de les diferències entre les mostres dels anys 2010 i 2016, suposant que ni les meganucleasses no han tingut cap efecte sobre els nivells de transposició detectables.

Es procedirà a generar la detecció de les transposicions en cadascun dels clons obtinguts a partir de totes les seqüenciacions mitjançant el *workflow* definit en el segon capítol d'aquest treball.

També es buscaran varies alternatives per comprovar si es milloren els resultats creant pools entre les mostres o bé per clons o per anys, si és possible millorar el grau de detecció d'aquests polimorfismes variant la llibreria de transposons, quin és el millor *coverage* per treballar les dades així també sí és millor crear pools combinant les *raw data* o bé les dades un cop alineades contra el genoma.

4.2.- Detecció de polimorfismes en els diferents genomes

Es procedeix a realitzar la detecció dels polimorfismes de transposons tal com s'ha anomenat prèviament per les diferents dades.

Les dades obtingudes per cadascuna de les seqüenciacions han estat les següents:

Taula 14: estadístiques de les diferents seqüenciacions

Mostra	*Coverage*	Nº Insercions de transposons	Nº de delecions	Mitjana de la longitud dels fragments (pb)	SD de la longitud dels fragments (pb)	Longitud dels *read*s (pb)	SD dels *read*s (pb)
V18_DA_B00H6BX	13.0172	0	6	373.51	83.83	95.59	12.89
V18_DA_B00H6BY	10.2042	0	5	382.48	86.39	95.64	12.87
V18_DA_B00H6BZ	10.702	0	6	382.56	86.61	95.6	12.93
V18_DA_B00H6C0	10.533	1	6	395.07	88.08	95.52	13.02
V18_DA_B00H6C1	11.133	1	6	365.42	84.13	95.73	12.74
V18_DA_B00H6C2	10.789	1	5	382.66	87.27	95.68	12.81
V18_DA_B00H6C3	11.146	4	8	347.71	73.83	97.24	11.91
V18_DA_B00H6C4	12.274	1	6	314	61.61	97.46	11.56
V18_DA_B00H6C6	11.849	1	4	344.85	73.48	97.22	11.92
V18_DA_B00H6C8	10.419	0	4	360.54	76.48	97.25	11.90
V18_DA_B00H6C9	5.39	0	5	331.27	69.69	97.38	11.55
V18_DA_B00H6CA	12.137	0	5	342.25	72.49	97.3	11.79

En general si ens fixem en les mostres del 2010 (X,Y,Z i 0) no veiem que hi hagi cap diferència significativa amb les insercions i delecions amb els altres períodes de temps així com tampoc amb els altres clons. Únicament destaca la mostra V18_DA_B00H6C3 que trobem 4 insercions i 8 delecions. Si ens fixem en les delecions i insercions però veiem que en molts casos ni tant sols coincideixen entre períodes (Veure annexe IX), l'única que es manté en 4 d'aquestes la trobem al cromosoma 13 entre les posicions 9821490 i 9822260.

A la taula disponible a l'annex VIII s'observa que molts dels polimorfismes d'inserció tenen una zigositat de -1. Ens pot donar informació de si una inserció és real o no. En individus heterozigots normalment es consideren com a insercions reals totes aquelles que siguin superiors a 0.5 en aquest cas al ser individus haploides considerem que com a mínim ha de tenir un valor de 0.75. Aquests valors de zigositat igual a -1 són degut a que no hi ha *softclipped reads* aquest tipus de *read*s són els que s'utilitzen per calcular la zigositat, quan no n'hi ha no es pot calcular i *jitterbug* retorna el valor de -1.

A partir de les dades obtingudes, no sembla que s'hagi produït cap moviment de transposons o en cas que se'n hagi produït siguin molts pocs, o bé que el *coverage* sigui molt baix. Així doncs, sembla que no s'ha produït un procés evolutiu durant aquest període de temps pels resultats obtinguts.

4.2.1.- Creació de *pools*

Es va procedir a desenvolupar *pools* de les dades obtingudes agrupant bé per any (2010 i 2016) o bé per clons (1 del 2010 i 2 del 2016). Això es realitza ja que pot ésser que tenim *coverages* massa baixos per detectar els polimorfismes de transposons.

Alhora d'agrupar tenim dues alternatives:

- Ajuntar els *fastq* en un únic *fastq* i realitzar tot el procés de filtratge i realitzar la detecció

- Ajuntar els *bam files* dels clons corresponents i realitzar la detecció de polimorfismes

En principi s'espera el mateix resultat, però cal comprovar-ho experimentalment. Es va procedir a crear els pools agrupant els clons del any 2010 i comparant-los entre sí amb les dues aproximacions:

Per tal de crear el *pool* de dades es van agrupar els diferents *fastq* senzillament concatenant els 8 *fastq*, els 4 *paired-end forward* i els 4 *paired-end reverse*. Es van concatenar en l'ordre següent: B00H6BX, B00H6BY, B00H6BZ, B00H6C0. Un cop creat el *pool* senzillament es va seguir la *pipeline* descrita fins a obtenir els resultats següents pel que fa a les insercions:

Taula 15: Detecció de polimorfismes d'inserció combinant *fastq*

2010 TE									
chr	**polymorphis**	**program**	**start**	**end**	**.**	**sense**	**.**	**description**	**zygosity**
Chr04	jitterbug	TE_insertion	19982104	19982636	.	.	.	supporting_fv	0,001
Chr13	jitterbug	TE_insertion	9821494	9822260	.	.	.	supporting_fv	1
Chr14	jitterbug	TE_insertion	9866381	9867018	.	.	.	supporting_fv	0,5
Chr17	jitterbug	TE_insertion	3905296	3905736	.	.	.	supporting_fv	1
Chr19	jitterbug	TE_insertion	13162713	13163064	.	.	.	supporting_fv	1

Observem que finalment després de filtrar s'obtenen insercions detectades un total de 4 en la mostra de l 'any 2010. S'augmenta un increment en la detecció respecte els clons individuals en què no s'havia detectat cap inserció

Pel que fa a la quantitat de *reads* recuperats es van obtenir els resultats descrits en la següent figura generada mitjançant *samtools flagstat* pel que fa als resultats obtinguts mesclant *fastq:*

```
crag@crag-optiplex-990:~/CRAG_pol/Physco_NGS/output_BAM/2010/merged_2010_bam/merged_2010_fastq$
samtools flagstat merged2010fastq.r1-r2.sw.sam.r1-r2.sw.psorted.aln.newtag.bam
220096870 + 0 in total (QC-passed reads + QC-failed reads)
0 + 0 duplicates
141155241 + 0 mapped (64.13%:-nan%)
220096870 + 0 paired in sequencing
110048435 + 0 read1
110048435 + 0 read2
139620036 + 0 properly paired (63.44%:-nan%)
140015613 + 0 with itself and mate mapped
1139628 + 0 singletons (0.52%:-nan%)
329227 + 0 with mate mapped to a different chr
190205 + 0 with mate mapped to a different chr (mapQ>=5)
```

Figura 23: Codi utilitzat per obtenir l'anàlisis de la combinació de *fastq* mitjançant flagstat

Òbviament, s'observa un increment de *reads, discordand reads i singletons* indicador de que el *pool* s'ha creat correctament.

Per altra banda es van combinar els diferents *bam files,* això es va fer mitjançant el programa *samtools merge* que combina varis arxius de tipus sam o *bam*. Es va obtenir exactament els mateixos resultats pel que fa a la predicció de polimorfismes que ajuntant els *fastq*. Els resultats obtinguts per la predicció de delecions també va ser exactament la mateixa, que és la descrita en la següent taula:

Taula 16: Delecions detectades per la combinació de totes les mostres del 2010 tant combinant *bam* files com *fastq*

2010 pindel coverage 44.41									
									LENGTH
Chr01	TE_deletion	pindel	24883958	24886674	.	+	.	score=7;total	2716
Chr04	TE_deletion	pindel	11564700	11565566	.	+	.	score=10;tot	866
Chr06	TE_deletion	pindel	3184305	3184666	.	+	.	score=5;total	361
Chr07	TE_deletion	pindel	3172773	3189677	.	+	.	score=5;total	16904
Chr14	TE_deletion	pindel	13278064	13290238	.	+	.	score=5;total	12174
Chr19	TE_deletion	pindel	5958069	5975760	.	+	.	score=8;total	17691
Chr20	TE_deletion	pindel	3985087	3985823	.	+	.	score=6;total	736
Chr21	TE_deletion	pindel	3465992	3473451	.	+	.	score=32;tot	7459
Chr21	TE_deletion	pindel	3473588	3477337	.	+	.	score=386;to	3749

S'obtenen en total 9 delecions de transposons tant combinant els *bam files* com els *fastq* així com 2 insercions de transposons. La quantitat d'insercions i delecions detectades és exactament la mateixa que combinant els *fastq*, obtenint 9 delecions i 5 insercions. Sembla que la creació de *pools* en diferents estadis de l'estudi no altera els resultats. Observem també pràcticament els mateixos valors mitjançant el programa *samtools flagstat:*

```
crag@crag-optiplex-990:~/CRAG_pol/Physco_NGS/output_BAM/2010/merged_2010_bam/merged_2010_bam$
samtools flagstat merged2010_psorted.bam
220096870 + 0 in total (QC-passed reads + QC-failed reads)
0 + 0 duplicates
141155242 + 0 mapped (64.13%:-nan%)
220096870 + 0 paired in sequencing
110048435 + 0 read1
110048435 + 0 read2
```

```
139619936 + 0 properly paired (63.44%:-nan%)
140015615 + 0 with itself and mate mapped
1139627 + 0 singletons (0.52%:-nan%)
329373 + 0 with mate mapped to a  iferent chr
190243 + 0 with mate mapped to a  iferent chr (mapQ>=5)
```

Figura 24: *output de flagstat mesclant bam* s'observa que es recuperen pràcticament els mateixos *read*s que mesclant els *fastq*

Ja que les diferències són mínimes obtenint exactament els mateixos resultats s'opta per combinar els *bam files* ja que s'havien generat tots amb anterioritat. També cal destacar que sembla que la creació de *pools* augmentant el *coverage* permet detectar més polimorfismes tals que en *coverages* més baixos no era possible amb uns nivells de zigositat alts.

Es combinen inicialment els *bam files* de les 4 seqüenciacions del 2010 i 4 del 2016 clon 1 i 4 del 2016 corresponent al clon 2, així com una darrer *pool* agrupant totes les mostres del 2016. Els resultats són els següents:

Taula 17: Resultats obtinguts de la creació de *pools* de les diferents mostres. En color morat hi ha aquells polimorfismes comunes entre el període de temps 2010 i 2016

2010 pindel coverage 44.41									
									LENGTH
Chr01	TE_deletion	pindel	24883958	24886674	.	+	.	score=7;total	2716
Chr04	TE_deletion	pindel	11564700	11565566	.	+	.	score=10;tot	866
Chr06	TE_deletion	pindel	3184305	3184666	.	+	.	score=5;total	361
Chr07	TE_deletion	pindel	3172773	3189677	.	+	.	score=5;total	16904
Chr14	TE_deletion	pindel	13278064	13290238	.	+	.	score=5;tota	12174
Chr19	TE_deletion	pindel	5958069	5975760	.	+	.	score=8;total	17691
Chr20	TE_deletion	pindel	3985087	3985823	.	+	.	score=6;tota	736
Chr21	TE_deletion	pindel	3465992	3473451	.	+	.	score=32;tot	7459
Chr21	TE_deletion	pindel	3473588	3477337	.	+	.	score=386;to	3749
2016_1 pindel coverage 39.035									
									LENGTH
Chr01	TE_deletion	pindel	24883958	24886674	.	+	.	score=44;tot	2716
Chr07	TE_deletion	pindel	3172773	3189677	.	+	.	score=5;total	16904
Chr18	TE_deletion	pindel	2306810	2328228	.	+	.	score=4;tota	21418
Chr21	TE_deletion	pindel	3465992	3473451	.	+	.	score=22;tot	7459
Chr21	TE_deletion	pindel	3473588	3477337	.	+	.	score=268;to	3749
Chr23	TE_deletion	pindel	13983880	13990050	.	+	.	score=6;tota	6170
2016_2 pindel coverage 45.999									
									LENGTH
Chr01	TE_deletion	pindel	22129517	22135957	.	+	.	score=4;tota	6440
Chr01	TE_deletion	pindel	24883958	24886674	.	+	.	score=10;tot	2716
Chr02	TE_deletion	pindel	18675023	18681569	.	+	.	score=6;tota	6546
Chr07	TE_deletion	pindel	3172773	3189677	.	+	.	score=4;total	16904
Chr08	TE_deletion	pindel	17562286	17568641	.	+	.	score=9;tota	6355
Chr18	TE_deletion	pindel	8249891	8259898	.	+	.	score=6;tota	10007
Chr19	TE_deletion	pindel	5958069	5975760	.	+	.	score=15;tot	17691
Chr21	TE_deletion	pindel	3465992	3473451	.	+	.	score=18;tot	7459
Chr21	TE_deletion	pindel	3473588	3477337	.	+	.	score=289;to	3749
2016_total pindel coverage 85.12									
									LENGTH
Chr01	TE_deletion	pindel	18910634	18919030	.	+	.	score=4;tota	8396
Chr01	TE_deletion	pindel	22129517	22135957	.	+	.	score=6;tota	6440
Chr01	TE_deletion	pindel	24883958	24886674	.	+	.	score=80;tot	2716
Chr02	TE_deletion	pindel	18675023	18681569	.	+	.	score=6;tota	6546
Chr04	TE_deletion	pindel	11564700	11565566	.	+	.	score=14;tot	866
Chr06	TE_deletion	pindel	3184305	3184666	.	+	.	score=4;total	361
Chr07	TE_deletion	pindel	3172773	3189677	.	+	.	score=8;total	16904
Chr08	TE_deletion	pindel	17562286	17568641	.	+	.	score=16;tot	6355
Chr17	TE_deletion	pindel	7125549	7127566	.	+	.	score=6;tota	2017
Chr18	TE_deletion	pindel	2306810	2328228	.	+	.	score=12;tot	21418
Chr18	TE_deletion	pindel	8249891	8259898	.	+	.	score=8;tota	10007
Chr19	TE_deletion	pindel	5958069	5975760	.	+	.	score=30;tot	17691
Chr21	TE_deletion	pindel	3465992	3473451	.	+	.	score=39;tot	7459
Chr21	TE_deletion	pindel	3473588	3477337	.	+	.	score=556;to	3749
Chr23	TE_deletion	pindel	13983880	13990050	.	+	.	score=8;tota	6170
scaffold_38	TE_deletion	pindel	81145	95854	.	+	.	score=6;tota	14709
2010 jitterbug TE									
									ZIGOSITY
Chr04	jitterbug	TE_insertion	19982104	19982636	.	.	.	supporting_fi	0,01
Chr13	jitterbug	TE_insertion	9821494	9822260	.	.	.	supporting_fi	1
Chr14	jitterbug	TE_insertion	9866381	9867018	.	.	.	supporting_fi	0,5
Chr17	jitterbug	TE_insertion	3905296	3905705	.	.	.	supporting_fi	1
Chr19	jitterbug	TE_insertion	13162713	13163142	.	.	.	supporting_fi	1
2016_1 jitterbug TE									
									ZIGOSITY
Chr02	jitterbug	TE_insertion	15769990	15770300	.	.	.	supporting_fi	0,251
Chr02	jitterbug	TE_insertion	15770927	15771404	.	.	.	supporting_fi	0,753
Chr13	jitterbug	TE_insertion	9821520	9822223	.	.	.	supporting_fi	1
Chr14	jitterbug	TE_insertion	9866412	9867009	.	.	.	supporting_fi	0,44
Chr17	jitterbug	TE_insertion	3905191	3905704	.	.	.	supporting_fi	1
Chr19	jitterbug	TE_insertion	13162789	13163060	.	.	.	supporting_fi	1
2016_2 jitterbug TE									
									ZIGOSITY
Chr13	jitterbug	TE_insertion	9821564	9822223	.	.	.	supporting_fi	1
Chr17	jitterbug	TE_insertion	3905271	3905592	.	.	.	supporting_fi	1
2016_total jitterbug TE									
									ZIGOSITY
Chr03	jitterbug	TE_insertion	9649730	9650173	.	.	.	supporting_fi	-1
Chr05	jitterbug	TE_insertion	16791978	16792541	.	.	.	supporting_fi	1
Chr07	jitterbug	TE_insertion	2570139	2570496	.	.	.	supporting_fi	1
Chr11	jitterbug	TE_insertion	11776186	11776729	.	.	.	supporting_fi	1
Chr13	jitterbug	TE_insertion	9821564	9822223	.	.	.	supporting_fi	1
Chr14	jitterbug	TE_insertion	9866412	9866947	.	.	.	supporting_fi	0,55
Chr17	jitterbug	TE_insertion	3905271	3905592	.	.	.	supporting_fi	1
Chr19	jitterbug	TE_insertion	12868733	12869146	.	.	.	supporting_fi	-1
Chr19	jitterbug	TE_insertion	13162789	13163060	.	.	.	supporting_fi	1
Chr21	jitterbug	TE_insertion	3102823	3103261	.	.	.	supporting_fi	1
Chr22	jitterbug	TE_insertion	14314849	14315427	.	.	.	supporting_fi	1

Primerament observem que amb *coverage* molt elevats (superiors a 80 vegades la mida del genoma) s'incrementen molt els valors d'insercions i delecions detectats és el cas de la combinació dels 8 *bam files* de les seqüenciacions del 2016, en què observem 118 insercions i 16 delecions. Moltes d'aquestes insercions veiem però que no es detecten en les altres combinacions dels *bam files* del 2016, per una part pot ser que aquests siguin artefactes o bé que els *coverage* encara siguin baixos, això es contrastarà en el darrer apartat d'aquest capítol. En canvi, a *coverage* baixos no es detecten pràcticament polimorfismes com passa en el cas de les mostres individuals.

Taula 18: Polimorfismes detectats a diferents *coverage* per la mateixa mostra

	(COVERAGE 5)	**(COVERAGE 40)**	**(COVERAGE 80)**
Insercions de transposons	0	6	11
Delecions de transposons	2	6	16
Insercions elements RLG 1	1	29	107

S'ha observat que *coverages* excessivament elevats en la gran majoria de programes disminueix la precisió dels programes augmentant els falsos positius i tenint un nombre més elevat d'artefactes, això es pot comprovar en la figura 28 (Rishishwar *et al*., 2016):

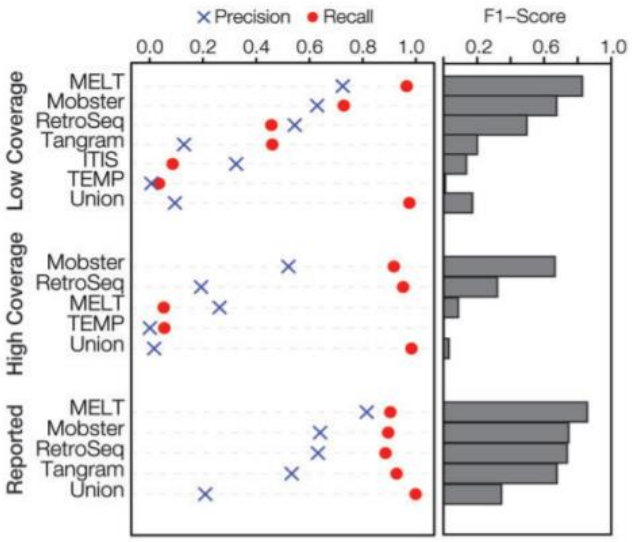

Figura 25: Precisió de diferents programes de detecció d'insercions de transposons. En general es veu que la sensiblitat disminueix en tots els programes quan el *coverage* és molt elevat (Rishishwar *et al*., 2016)

Per altra banda, pel que fa als resultats en sí s'observen que hi ha tot un seguit de polimorfismes que són comuns entre el 2010 i el 2016 (en lila a la taula 17) així com tot un seguit de polimorfismes que són únics del període 2016, són aquells de color blanc a les taules. Aquests serien els resultats que esperaríem si s'haguessin produït alguna recombinació produint-se una deleció durant aquest període de temps o s'hagués produït una inserció així els resultats semblen indicar que sí que s'ha produït un canvi evolutiu durant aquest període de temps.

No obstant també s'observen tot un seguit de polimorfismes durant el període del 2010 que no s'observen el 2016 aquests bé poden ser artefactes o bé insercions que no s'han detectat a posterior durant el 2016. Es van verificar aquests resultats visualitzant els *reads* mitjançant algun programa com *IGV* per tal de verificar si són artefactes o no i què és el que ha passat en cada cas.

4.2.2.- Detecció d'insercions dintre de transposons

Tal i com hem realitzat en els diferents ecotips i veient els resultats obtinguts (tenint una baixa quantitat de polimorfismes) es decideix llençar també únicament la llibreria d'elements *RLG1* amb *jitterbug*.

En l'annex XI hem pogut comprovar que s'han detectat tot un seguit de polimorfismes però en la majoria de casos no són molt nombrosos i hi ha algunes incongruències.

S'observa que es detecten moltes més insercions incrementant en alguns casos de 11 insercions a 107 insercions dintre d'elements *RLG1*. A més es detecten insercions úniques en el 2010, això no sembla possible que succeeixi ja que s'haurien d'haver mantingut el 2016 i no és el cas.

4.2.3.- Resultats

Tenint en compte únicament les insercions i delecions detectades llençant tota la llibreria d'elements de transposons obtenim els següents resultat:

Taula 19: Insercions i delecions detectades durant els diferents períodes de temps

	2010	2016 clon 1	2016 clon 2	2016 tots els clons
Insercions de transposons	5	6	2	11
Delecions de transposons	9	6	9	16
Insercions elements RLG 1	30	29	36	107

Si que s'observa la mateixa quantitat de delecions el 2010 i el 2016 però no és així amb el clon 1 o agrupant tots els clons del 2016. Tampoc no hi ha una lògica en quan a les insercions detectades.

A mòde de resum els resultats obtinguts són els següents:

Taula 20: Insercions i delecions detectades durant els diferents períodes de temps agrupades

	Únics 2010	**Únics 2016**	**Comuns entre 2010-2016**
Insercions de transposons	0	6	4
Delecions de transposons	2	9	7
Insercions elements RLG 1	5	29	9

S'observen polimorfismes únics durant el període 2010. Aquest fet no és lògic ja que al mantenir-se de forma vegetativa fins el 2016. Això passa amb dues delecions i 5 insercions. Sí que és lògic que s'observi 35 insercions i 9 delecions el 2016 que pot ser que s'hagin produït durant aquest període.

Els resultats obtinguts s'han filtrat tenint en compte que la zigositat ha de ser igual o superior a 0.75. Com hem dit es tracta de clons de *Physcomitrella patens* que es troben en estadi haploide, per aquest fet s'opta per filtrar amb una zigositat de com a mínim 0.75. Els resultats s'han agrupat en aquelles que són úniques el 2010, aquelles úniques del 2016 i aquelles que són en comú entre els diferents períodes de temps, el resultat de tots aquests polimorfismes són els següents:

Taula 21: total de polimorfismes detectats durant els diferents períodes de temps filtrats per zigositat

Common 2010-2016 TE									
chr	**polymorphism**	**program**	**start**	**end**	**.**	**sense**	**.**	**description**	**zygosity**
Chr13	jitterbug	TE_insertion	9821564	9822223	.	.	.	supporting_fwd_reads=21; supporting_rev_reads=2; cluster_p	1
Chr14	jitterbug	TE_insertion	9866412	9866947	.	.	.	supporting_fwd_reads=3; supporting_rev_reads=2; cluster_pa	1
Chr17	jitterbug	TE_insertion	3905296	3905592	.	.	.	supporting_fwd_reads=7; supporting_rev_reads=4; cluster_pa	1
Chr19	jitterbug	TE_insertion	13162789	13163060	.	..	supporting_fwd_reads=3; supporting_rev_reads=2; cluster_pair_ID=50; lib=		1
Uniq 2010 TE									
No one									
Uniq 2016 TE									
chr	**polymorphism**	**program**	**start**	**end**	**.**	**sense**	**.**	**description**	**zygosity**
Chr03	jitterbug	TE_insertion	9649730	9650173	.	.	.	supporting_fwd_reads=2; supporting_rev_reads=2; cluster_pa	1
Chr05	jitterbug	TE_insertion	16791978	16792541	.	..	supporting_fwd_reads=2; supporting_rev_reads=2; cluster_pair_ID=538; lib		1
Chr07	jitterbug	TE_insertion	2570139	2570496	.	.	.	supporting_fwd_reads=4; supporting_rev_reads=6; cluster_pa	1
Chr11	jitterbug	TE_insertion	11776186	11776729	.	..	supporting_fwd_reads=5; supporting_rev_reads=2; cluster_pair_ID=973; lib		1
Chr21	jitterbug	TE_insertion	3102823	3103261	.	.	.	supporting_fwd_reads=4; supporting_rev_reads=2; cluster_pa	1
Chr22	jitterbug	TE_insertion	14314849	14315427	.	..	supporting_fwd_reads=4; supporting_rev_reads=2; cluster_pair_ID=1750; li		1
Common 2010-2016 RLG1									
chr	**polymorphism**	**program**	**start**	**end**	**.**	**sense**	**.**	**description**	**zygosity**
Chr02	jitterbug	TE_insertion	11397000	11397515	.	..	supporting_fwd_reads=6; supporting_rev_reads=9; cluster_pair_ID=368; lib		1
Chr06	jitterbug	TE_insertion	13845451	13845791	.	..	supporting_fwd_reads=6; supporting_rev_reads=7; cluster_pair_ID=1170; li		1
Chr09	jitterbug	TE_insertion	15844736	15845152	.	..	supporting_fwd_reads=4; supporting_rev_reads=8; cluster_pair_ID=1661; li		1
Chr12	jitterbug	TE_insertion	16063071	16063649	.	..	supporting_fwd_reads=14; supporting_rev_reads=7; cluster_pair_ID=2120;		1
Chr13	jitterbug	TE_insertion	9821588	9821954	.	.	.	supporting_fwd_reads=11; supporting_rev_reads=6; cluster_p	1
Chr13	jitterbug	TE_insertion	16551164	16551602	.	..	supporting_fwd_reads=21; supporting_rev_reads=15; cluster_pair_ID=2292		1
Chr17	jitterbug	TE_insertion	3905296	3905592	.	.	.	supporting_fwd_reads=4; supporting_rev_reads=5; cluster_pa	1
Chr17	jitterbug	TE_insertion	15342811	15343318	.	..	supporting_fwd_reads=4; supporting_rev_reads=6; cluster_pair_ID=2857; li		1
Chr19	jitterbug	TE_insertion	15152628	15152829	.	..	supporting_fwd_reads=6; supporting_rev_reads=6; cluster_pair_ID=3150; li		1
Uniq 2010 RLG1									
chr	**polymorphism**	**program**	**start**	**end**	**.**	**sense**	**.**	**description**	**zygosity**
Chr02	jitterbug	TE_insertion	18079340	18079826	.	.	.	supporting_fwd_reads=6; supporting_rev_reads=7; cluster_pa	1
Chr07	jitterbug	TE_insertion	9720128	9720749	.	.	.	supporting_fwd_reads=6; supporting_rev_reads=2; cluster_pa	1
Chr11	jitterbug	TE_insertion	1844123	1844560	.	.	.	supporting_fwd_reads=5; supporting_rev_reads=8; cluster_pa	1
Chr14	jitterbug	TE_insertion	12145602	12146453	.	.	.	supporting_fwd_reads=3; supporting_rev_reads=4; cluster_pa	1
Chr21	jitterbug	TE_insertion	12851380	12851736	.	.	.	supporting_fwd_reads=6; supporting_rev_reads=11; cluster_p	1
Uniq 2016 RLG1									
chr	**polymorphism**	**program**	**start**	**end**	**.**	**sense**	**.**	**description**	**zygosity**
Chr02	jitterbug	TE_insertion	13227517	13227905	.	.	.	supporting_fwd_reads=2; supporting_rev_reads=2; cluster_pa	1
Chr09	jitterbug	TE_insertion	5429903	5430142	.	.	.	supporting_fwd_reads=29; supporting_rev_reads=10; cluster_	1
Chr11	jitterbug	TE_insertion	9536674	9536919	.	.	.	supporting_fwd_reads=4; supporting_rev_reads=4; cluster_pa	1
Chr13	jitterbug	TE_insertion	14880495	14880899	.	.	.	supporting_fwd_reads=3; supporting_rev_reads=7; cluster_pa	1
Chr14	jitterbug	TE_insertion	13650837	13651150	.	.	.	supporting_fwd_reads=2; supporting_rev_reads=2; cluster_pa	1
Chr15	jitterbug	TE_insertion	2185152	2185587	.	.	.	supporting_fwd_reads=2; supporting_rev_reads=6; cluster_pa	1
Chr15	jitterbug	TE_insertion	10630018	10630564	.	.	.	supporting_fwd_reads=4; supporting_rev_reads=3; cluster_pa	1
Chr18	jitterbug	TE_insertion	10139585	10140086	.	.	.	supporting_fwd_reads=2; supporting_rev_reads=3; cluster_pa	1
Chr18	jitterbug	TE_insertion	12213267	12213946	.	.	.	supporting_fwd_reads=2; supporting_rev_reads=2; cluster_pa	1
Chr19	jitterbug	TE_insertion	12419174	12419576	.	.	.	supporting_fwd_reads=11; supporting_rev_reads=4; cluster_p	1
Chr19	jitterbug	TE_insertion	13432192	13432553	.	.	.	supporting_fwd_reads=4; supporting_rev_reads=2; cluster_pa	1
Chr20	jitterbug	TE_insertion	6770242	6770521	.	.	.	supporting_fwd_reads=12; supporting_rev_reads=27; cluster_	1
Chr22	jitterbug	TE_insertion	10986210	10986737	.	.	.	supporting_fwd_reads=4; supporting_rev_reads=3; cluster_pa	1
Chr24	jitterbug	TE_insertion	1972456	1972702	.	.	.	supporting_fwd_reads=3; supporting_rev_reads=4; cluster_pa	1
Common 2010-2016 pindel									
chr	**polymorphism**	**program**	**start**	**end**	**.**	**sense**	**.**	**description**	**length**
Chr01	TE_deletion	pindel	24883958	24886674	.	+	.	score=7;total_reads_uniq=6;fwd_reads_uniq=0;rev_reads_uni	2716
Chr04	TE_deletion	pindel	11564700	11565566	.	+	.	score=10;total_reads_uniq=5;fwd_reads_uniq=1;rev_reads_u	866
Chr06	TE_deletion	pindel	3184305	3184666	.	+	.	score=5;total_reads_uniq=4;fwd_reads_uniq=4;rev_reads_uni	361
Chr07	TE_deletion	pindel	3172773	3189677	.	+	.	score=5;total_reads_uniq=4;fwd_reads_uniq=0;rev_reads_uni	16904
Chr19	TE_deletion	pindel	5958069	5975760	.	+	.	score=8;total_reads_uniq=4;fwd_reads_uniq=3;rev_reads_uni	17691
Chr21	TE_deletion	pindel	3465992	3473451	.	+	.	score=32;total_reads_uniq=18;fwd_reads_uniq=0;rev_reads_	7459
Chr21	TE_deletion	pindel	3473588	3477337	.	+	.	score=386;total_reads_uniq=14;fwd_reads_uniq=0;rev_reads_	3749
Uniq 2010 pindel									
chr	**polymorphism**	**program**	**start**	**end**	**.**	**sense**	**.**	**description**	**length**
Chr14	TE_deletion	pindel	13278064	13290238	.	+	.	score=5;total_reads_uniq=4;fwd_reads_uniq=4;rev_reads_uni	12174
Chr20	TE_deletion	pindel	3985087	3985823	.	+	.	score=6;total_reads_uniq=3;fwd_reads_uniq=2;rev_reads_uni	736
Uniq 2016 pindel									
chr	**polymorphism**	**program**	**start**	**end**	**.**	**sense**	**.**	**length**	**length**
Chr01	TE_deletion	pindel	18910634	18919030	.	+	.	score=4;total_reads_uniq=3;fwd_reads_uniq=0;rev_reads_uni	8396
Chr01	TE_deletion	pindel	22129517	22135957	.	+	.	score=6;total_reads_uniq=4;fwd_reads_uniq=0;rev_reads_uni	6440
Chr02	TE_deletion	pindel	18675023	18681569	.	+	.	score=6;total_reads_uniq=3;fwd_reads_uniq=2;rev_reads_uni	6546
Chr08	TE_deletion	pindel	17562286	17568641	.	+	.	score=16;total_reads_uniq=6;fwd_reads_uniq=3;rev_reads_u	6355
Chr17	TE_deletion	pindel	7125549	7127566	.	+	.	score=6;total_reads_uniq=3;fwd_reads_uniq=1;rev_reads_uni	2017
Chr18	TE_deletion	pindel	2306810	2328228	.	+	.	score=12;total_reads_uniq=5;fwd_reads_uniq=4;rev_reads_u	21418
Chr18	TE_deletion	pindel	8249891	8259898	.	+	.	score=8;total_reads_uniq=3;fwd_reads_uniq=1;rev_reads_uni	10007
Chr23	TE_deletion	pindel	13983880	13990050	.	+	.	score=8;total_reads_uniq=4;fwd_reads_uniq=1;rev_reads_uni	6170
scaffold_38	TE_deletion	pindel	81145	95854	.	+	.	score=6;total_reads_uniq=3;fwd_reads_uniq=2;rev_reads_uni	14709

Observem en general que han augmentat molt el nombre de *discordand reads* per cadascuna de les insercions. Aquestes dades són les que ajuden a verificar si són insercions reals o bé artefactes incrementant de 2 a 1 en els casos individuals a 4,6 o 8 *discordand reads* per cadascun dels polimorfismes descrits.

De nou, els resultats semblen prou concloents. Sembla que hi ha tot un seguit de polimorfismes que es detecten de forma comuna el 2010 i el 2016 així com tot un seguit únics dels 2016, tot i així es continuen mantenint un seguit d'insercions i delecions úniques del 2010 fet que és estrany, caldrà verificar els 71 polimorfismes per tal de verificar si són o no són polimorfismes o bé són artefactes.

4.3.- Detecció d'artefactes en les seqüències, validació de les dades

Per tal de verificar si són o no són polimorfismes es van realitzar varis processos.
Per una part verificar si les delecions són artefactes o delecions reals, per tal de verificar-ho inicialment es mirarà si la seqüència en el genoma de referència són nucleòtids definits o bé són nucleòtids no definits (N) i per tant la seqüència no és present ja en el genoma de referència.

La comprovació de la resta de polimorfismes es realitzarà de forma manual contrast amb un *bam file* disponible de l'ecotip *Villersexel*. Això ens permetrà detectar tots aquells polimorfismes que són artefactes i que són errors en la seqüència de referència. Això és degut a que no esperem cap polimorfisme comú entre un ecotip (que va divergir evolutivament fa molt temps) amb els que s'hagin produït del 2008 a l'actualitat. Les possibilitats de que passi això són pràcticament nul·les per això es definiran aquestes situacions com a artefactes.

4.3.1.- Detecció d'artefactes: Seqüències no definides

En aquest si la seqüència no està definida en el genoma de referència i en aquesta regió s'ha produït un artefacte, no hi hauran *reads* que hagin alineat en aquesta regió. Aquesta situacions es poden produir en algunes delecions.

Es detecten dos d'aquestes situacions, corresponents a delecions artefactuals ja que la seqüència són N (és a dir nucleòtids no definits). Aquests els trobem en les següents situacions:

Taula 22: Delecions artefactuals detectades

Chromosome	polymorphism	program	Start	End	Length
Chr01	TE_deletion	*pindel*	24883958	24886674	2716
Chr04	TE_deletion	*pindel*	11564700	11565566	866

Totes les delecions s'han visualitzat a Ugene (Konechnikov *et al.*, 2012). Únicament s'han detectat dos artefactes ambdós són delecions comunes entre els períodes 2010 i 2016. La visualització a Ugene a estat la següent:

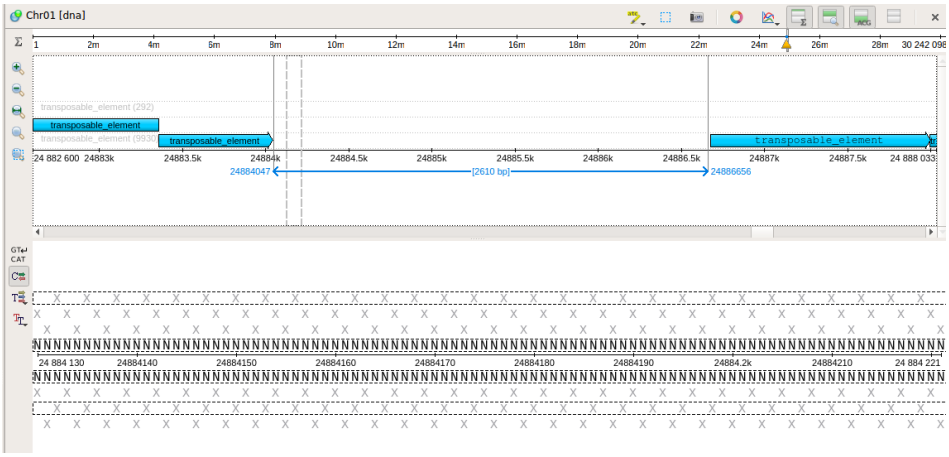

Figura 26: Detecció d'un artefacte en el cromosoma 1 en les posicions descrites anteriorment

4.3.2.- Detecció d'artefactes: Polimorfismes no detectats per baix *coverage*

Tal i com hem vist hi ha tot un seguit d'insercions detectades únicament el 2010 així com el 2016. Queda el dubte de si aquestes insercions no es mesuren per *coverage* baix a les mostres del 2010. Per tal de comprovar-ho es va visualitzar a IGV (Thorvaldsdóttir *et al.,* 2013). Aquest permet visualitzar els *reads* contra el genoma. Si es visualitza exactament el mateix tant el 2010 com el 2016 s'espera que realment siguin comuns encara que únicament s'hagin detectat durant un període de temps.

Es van visualitzar tots els polimorfismes mitjançant *IGV* (Thorvaldsdóttir *et al.,* 2013). Curiosament s'observa en tots els casos que els polimorfismes únics del 2010 també són presents el 2010 com és el de la figura 31:

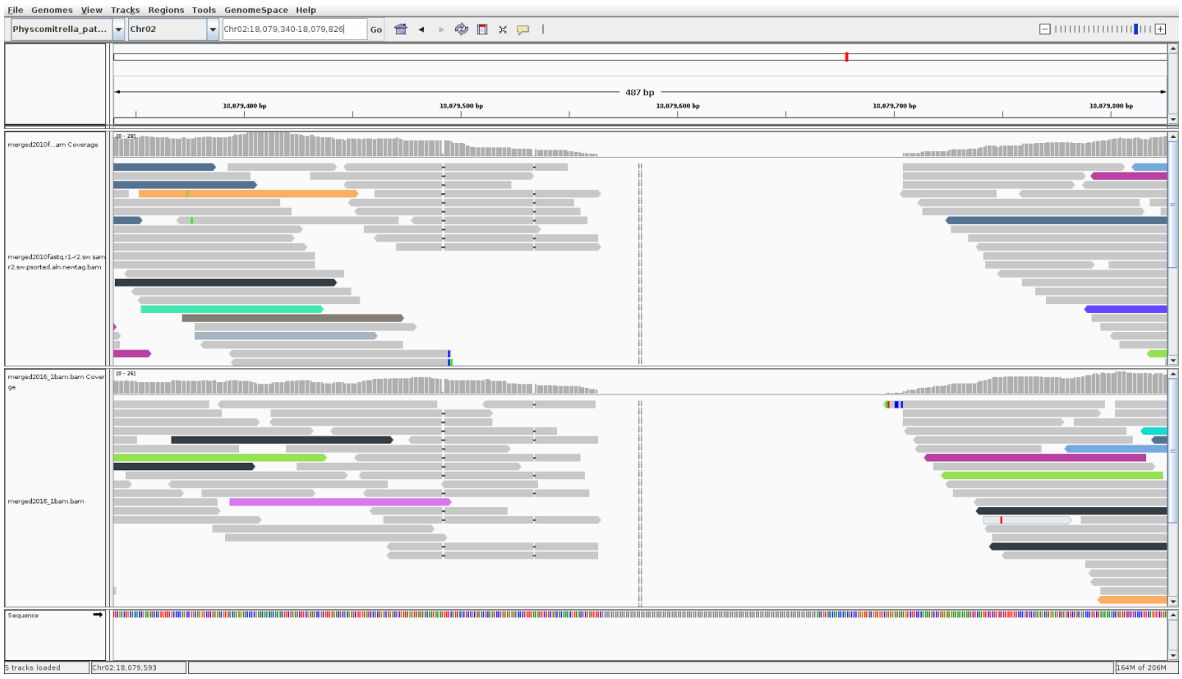

Figura 27: Visualització de les insercions a la part superior es visualitzen els *read*s del 2010 a la inferior del 2016

En aquesta figura es visualitza la inserció única durant el 2010 al cromosoma 2 entre les posicions 18.079.340 i 18.079.826. Tal i com es veu, es visualitza exactament els mateixos *read*s el 2010 i el 2016 però únicament s'ha detectat la inserció el 2010. També sorprèn que hi ha tota una regió de 200 pb aproximadament que no hi ha *reads* això fa pensar que pot ser un artefacte la pròpia inserció. Això passa per totes les insercions úniques del 2010.

A continuació es va procedir a realitzar el mateix per les insercions úniques del 2016 passant exactament el mateix, on es visualitza que són comunes entre el període 2010 i 2016, a sota hi ha un exemple d'una inserció d'un element *RLG1* detectada el 2016 i no el 2010 tot i que el que es visualitza és la mateixa situació:

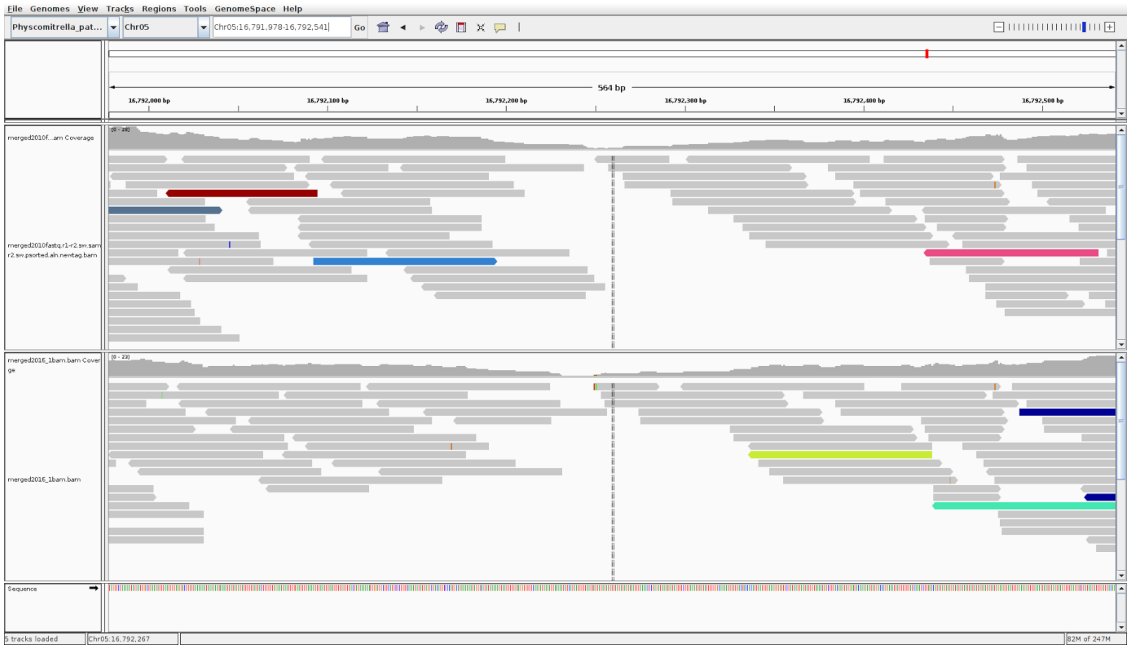

Figura 28: Inserció d'un element *RLG1* detectada el 2016 al cromosoma 5 entre les posicions 16791978 i 16792541

No es detecten polimorfismes únics ni en el període 2010 ni 2016. A més a més passen varies situacions estranyes:

- Per un costat, tot i tenir calculada una zigositat = 1 en molts casos els *discordand reads* estan en zones en què hi ha una gran quantitat de *reads* correctament mapejats. Això no és normal en aquest cas ja que esperaríem únicament una zona on hi hagi una gran quantitat de *discordant reads* i pocs *reads* mapejats correctament.

- Per altra banda, en moltes d'aquestes insercions es detecten *discordand reads* que mapegen al mateix cromosoma i en posicions no molt allunyades d'on s'ha predit que és la posició del *read*, a uns 150-200 pb.

- Finalment, en molts casos en què s'ha predit una inserció flanquejant els *discordand reads* hi ha tota una gran quantitat de seqüència que no hi ha cap *read* alineats, fent pensar que en aquesta regió hi hauria d'haver una deleció i no pas una inserció detectada.

Un exemple d'això és la següent inserció detectada tant el 2010 com el 2016:

Figura 29: Inserció predita tant el 2010 com el 2016 on s'assembla molt més a una deleció que no a una inserció

Una inserció real d'un element transposable pren la següent forma:

Figura 30: inserció predita a *Villersexel*

I en cap cas no es prediu cap inserció d'aquest tipus en les dades, tot això fa porta a la conclusió que les insercions no són reals sinó que són artefactes. Per tal de verificar si és així es compararà amb la seqüenciació de l'ecotip *Villersexel* aquestes insercions.

4.3.3.- Detecció d'artefactes: Artefactes detectats contrastant amb l'ecotip *Villersexel*

Tal i com s'ha dit es compara els *reads* de les seqüenciacions dels diferents períodes de temps on s'ha detectat un possible polimorfisme amb l'ecotip *Villersexel*. No esperem que hi hagi cap inserció en comuna així que és un bon mètode per detectar si són o no són artefactes. Visualitzem una a una els polimorfismes, observant el següent per exemple en el cromosoma 1 a les posicions 24883958 a 24886674, aquest artefacte és molt similar al visualitzat en l'anterior apartat. Això és degut a que la gran majoria prenen aquesta forma.

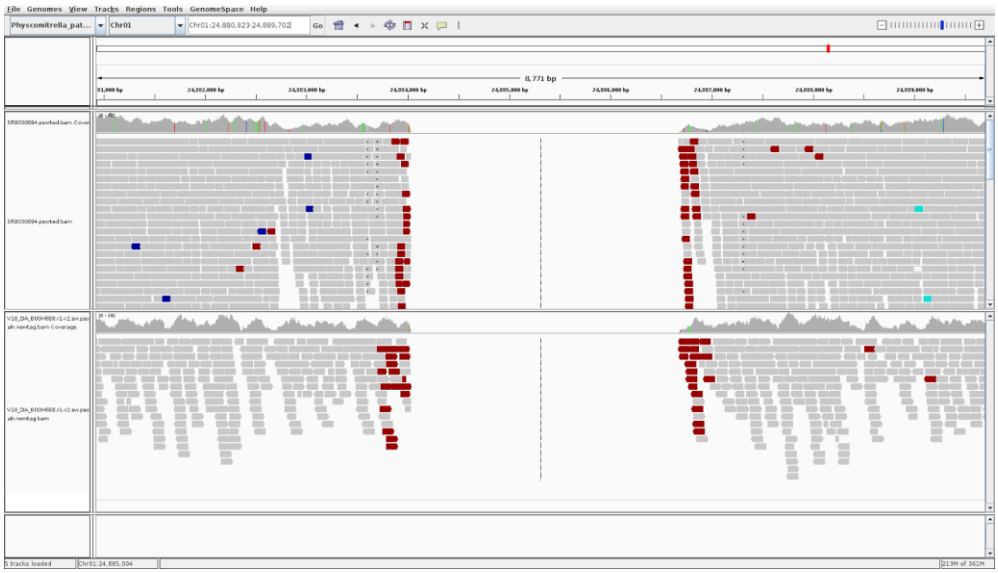

Figura 31: inserció comuna entre el 2010 i el 2016 , a baix *reads* de la seqüenciació del 2010 i a dalt *reads* de l'ecotip de *Physcomitrella patens Villersexel*

Veiem que es visualitza la mateixa situació tant en *Villersexel* com en les noves seqüenciacions. Passa exactament el mateix amb tota la resta dels polimorfismes tant insercions com delecions. Totes aquests polimorfismes detectats tant en *Villersexel* com en les noves seqüenciacions del 2010 i 2016 porten a pensar que es tracten d'errors en l'assemblatge del genoma de referència.

En resum, sembla que no hi ha cap polimorfisme real durant aquest període de temps del 2008 al 2016.

4.4.- Resum dels resultats

Després de tot el procés realitzat no s'ha detectat cap polimorfisme entre el període de temps 2008 a 2016. Tot i que, s'ha detectat algun polimorfisme després s'ha verificat que tots aquests polimorfismes són realment artefactes i concretament són errors en l'assemblatge del genoma de referència.

S'ha aconseguit llistar una gran quantitat d'errors que en tot cas serviran per millorar l'anotació i ensemblatge del genoma.

Sí que el procés realitzat ens ha permès saber quin és el millor *coverage* en el que treballar per tal de detectar els polimorfismes en *Physcomitrella patens* en un genoma altament repetitiu. Hem vist que treballar a *coverage* molt baixos no permet que es detectin molts d'aquests polimorfismes però per altra banda treballar a *coverage* molt elevats incrementa la detecció però a canvi disminueix molt la sensibilitat. Podem arribar a la conclusió que el millor *coverage* amb el que es pot treballar és entre 20 i 40 vegades la mida del genoma, amb concordança amb l'estudi (Rishishwar *et al.,* 2016) .

Per acabar, cal destacar que pel que fa a l'evolució el genoma de *P. patens* no hem pogut detectar cap increment en el nombre de polimorfismes. Encara queda però verificar si els *SNPs* observats durant aquest període de temps són artefactes o bé s'ha produït un procés evolutiu durant aquests darrers anys.

Aquest estudi l'està realitzant el grup en *DNA repair and genome engineering* que va realitzar la seqüenciació i s'està esperant els resultats d'aquestes anàlisis a l'hora de finalitzar aquest treball.

4.5.- Validació experimental dels resultats

Per tal de verificar la hipòtesi es van seleccionar dos d'aquests artefactes per tal de comprovar si realment són o no són artefactes. Es van dissenyar *primers* flanquejant aquest "polimorfisme" i es van realitzar *PCR*.

El primer artefacte seleccionat és un polimorfisme d'un element *RLG1* concretament el descrit en el cromosoma 19 entre les posicions 12.419.174 i 12.419.576.

Primerament es va visualitzar a IGV la zona obtenint el següent resultat:

Figura 32: artefacte seleccionat per la anàlisis a dalt en Gransden i a sota en *Villersexel*

Es van seleccionar oligonucleòtids flanquejant la zona per tal d'amplificar mitjançant *PCR*. Es va desenvolupar amb l'aplicació Primer3[5] :

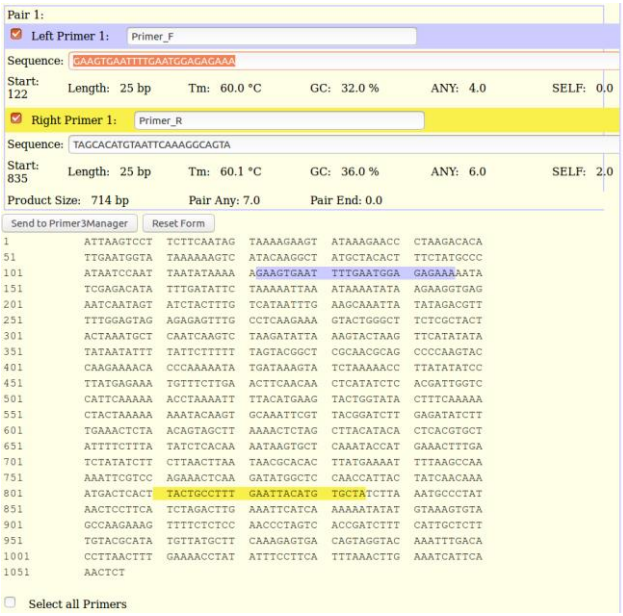

Figura 33: Selecció dels primers per ampliar la regió d'interès

Amb els primers seleccionats es va procedir primerament a visualitzar amb UGENE qui és el product esperat.

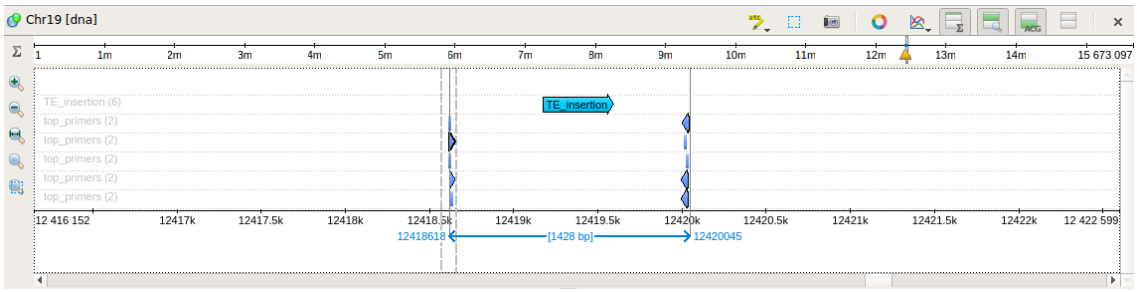

Figura 34: Producte esperat en cas que no hi hagi cap inserció. El producte és de 1428pb

A continuació, es va buscar els possibles efectes *offtarget* dels *primers* cercant si podien amplificar a altres bandes del genoma (cal considerar-ho ja que estem ampliant regions molt repetitives).Això es va fer mitjançant un blast local):

```
crag@crag-optiplex-990:~/CRAG_pol/Physco/sequencies/DATA_RefGenome/genome/Ppa_v3.0$ blastn -task
blastn-short -db Physcomitrella_patens.main_genome.scaffolds.fasta -query primers.fa -outfmt 6 -
max_target_seqs 1000 -word_size 23 | more

1       Chr19   100.00  25      0       0       1       25      12419028124190524e-06   50.1
```

Figura 35:Codi utilitzat per cercar possibles offtargets mitjançant BLASTn short

Veiem que s'espera un producte únic de 1428 pb en cas de que no sigui un artefacte. Les seqüències dissenyades són les següents:

- 5': GAAGTGAATTTTGAATGGAGAGAAA

-3': TAGCACATGTAATTCAAAGGCAGTA

Aquestes es van encarregar i es va procedir a realitzar les *PCR* sobre Gransden i *Villersexel* obtenint els següents resultats:

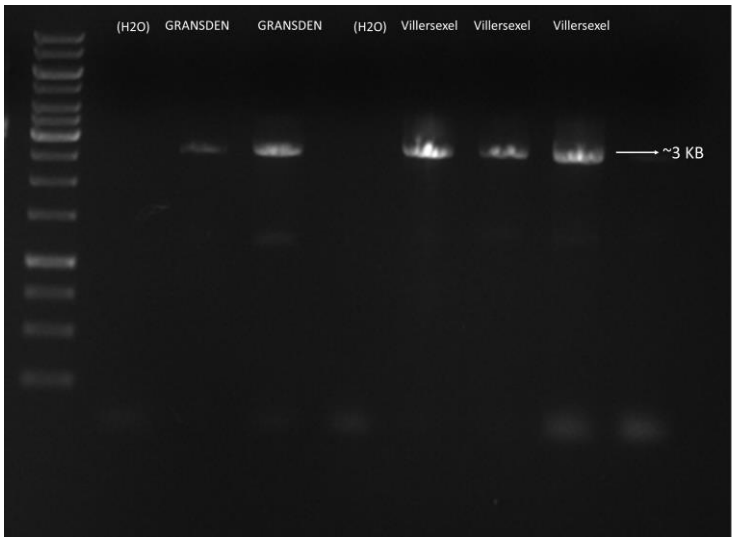

Figura 36: productes de la *PCR* visualitzades en gel d'electoforèsis

Veiem que únicament ha amplificat una banda de 3 KB aproximadament tant en Gransden com en *Villersexel*. Queda comprovat que són artefactes ja que no s'obté el patró de bandes esperats de 1400 kb i que la regió està mal assemblada. De totes formes es va enviar a seqüenciar el producte de *PCR* però al finalitzar aquest treball encara no es tenen els resultats. Es va repetir el mateix per un altra artefacte aquest queda descrit en l'annex X.

5. Conclusions

A partir de tot el treball podem extreure les següents conclusions:

- S'ha aconseguit desenvolupar un *workflow* fiable per tal de detectar el màxim nombre possible de polimorfismes causats per transposons a partir de seqüenciacions en format *paired-end* i de forma fiable.

- Hem pogut observar com el *coverage* és clau en els programes de detecció d'insercions:

  - Treballant a *coverages* baixos no és possible detectar gran part dels polimorfismes
  - Treballant a *coverages alts* disminueix en gran mesura la fiabilitat detectant un gran nombre d'artefactes

- S'ha aconseguit desenvolupar un mètode per forçar la detecció d'insercions dintre de transposons, útil per la detecció en genomes molt repetitius.

- No s'ha pogut detectar un canvi significatiu pel que fa als polimorfismes de transposons entre el període comprès entre el 2010 i 2016

- S'han detectat errors en l'assemblatge del genoma de referència gràcies a les reseqüenciacions dutes a terme durant els períodes 2010 i 2016, així com en els diferents ecotips.

- S'ha incrementat la quantitat de polimorfismes d'inserció de transposons detectats en l'ecotip *Villersexel*, detectant un gran nombre nou d'insercions d'elements *RLG1* a l'interior d'altres elements transposables

- S'ha generat la detecció de polimorfismes de transposons de l'ecotip *Reute*, detectant un nombre relativament baix de polimorfismes comparat amb l'altre ecotip.

- Els polimorfismes de transposons detectats apunten a que hi ha molta més distància evolutiva entre l'ecotip *Villersexel* i *Gransden* que entre *Reute* i *Gransden.* Això és degut a que es detecta una quantitat molt superior de polimorfismes en l'ecotip *Villersexel* que en l'ecotip *Reute.*

- Hem pogut validar que els elements *RLG1* són molt actius dintre el genoma de *Physcomitrella patens*, veient un alt nombre de polimorfismes causats per aquest element. Tant en el ecotip *Villersexel* com en l'ecotip *Reute.*

- S'ha observat com els elements *RLG1* tendeixen majoritàriament inserir-se preferencialment dintre d'altres elements transposables, detectant un alt nombre de polimorfismes en què es produeix aquesta situació

Tot aquest treball ha permès aplicar de forma pràctica els coneixements apresos durant el màster, sobretot, tots aquells que impliquen bioinformàtica, no podent aplicar l'àmbit més bioestadística del màster. Bàsicament ha estat enfocat en l'àmbit de l'estudi i processament de dades òmiques, podent practicar molts dels coneixements adquirits durant el transcurs d'aquest màster i ampliar coneixements teòrics sobre la bioquímica i biologia molecular dels elements transposables en el genoma.

Pel que fa al compliment dels objectius, moltes de les conclusions a les que s'ha arribat en aquest treball són la resposta directe als objectius plantejats. S'ha aconseguit avaluar una petita varietat d'eines de la gran quantitat d'eines que hi ha disponibles i que continuen en constant desenvolupament. Tot i que el conjunt d'eines utilitzades no són molt elevades sí que aquestes permeten arribar a l'objectiu final que és la detecció de polimorfismes de transposons d'una forma adequada amb les dades disponibles. També en aquest treball s'ha vist la importància de la comprovació de les dades. Sinó s'hagués fet una validació, tant *in silico* com *in vitro,* la conclusió a la que s'hauria arribat seria totalment contrària i, més important, errònia.

Malauradament no s'han pogut detectar polimorfismes de transposons durant el període de temps 2008-2016. Això pot ser degut a que s'ha mantingut per propagació vegetativa o expansió clonal i únicament hi ha hagut reproducció sexual un cop a l'any. També pot ser que senzillament no ha passat prou temps perquè podem detectar cap polimorfisme.

Tot i això, la informació generada ha estat útil en primer lloc per detectar errors en el genoma de referència (problemes en l'assemblatge del genoma). Aquests artefactes es troben en regions altament repetitives en mig de transposons que poden ser la causa del mal assemblatge en el genoma de referència.

Aquestes dades també han estat útils per realitzar proves amb els programes de detecció de transposons que ens han permès veure la importància del *coverage* per exemple en la detecció dels polimorfismes de transposons.

Hi ha un parell d'objectius, però, que no s'han pogut complir. El primer per causes externes ha estat la comparació amb l'estudi de *SNP*s durant aquest període de temps, això és degut a que l'estudi era realitzat per un altra grup independentment de la meva tasca encara restant que arribin els resultats. Per altra banda, no s'ha pogut realitzar la comparació dels programes d'insercions de transposons de forma experimental bé per incompatibilitat de les dades o que els programes tenien altres usos com la detecció de transposons polimòrfics en poblacions.

Per altra banda, hi ha altres objectius que està previst realitzar en un futur. Com per exemple la comparació amb les dades de SNPs es realitzarà en un futur proper quan aquestes dades s'hagin generat. També es comprovarà l'impacte dels transposons sobre gens propers en un futur pròxim mitjançant les seqüenciacions dels diferents ecotips obtingudes. Aquest és un procés més costós pel qual caldria analitzar diferents dades de *RNA-seq* en diferents teixits així com realitzar *RT-PCR*, aquest requereix una gran capacitat de temps que no s'ha pogut realitzar en aquest projecte.

Pel que fa a la planificació hi ha hagut varies modificacions de les inicialment plantejades. Inicialment es va plantejar el següent diagrama de *Gannt*:

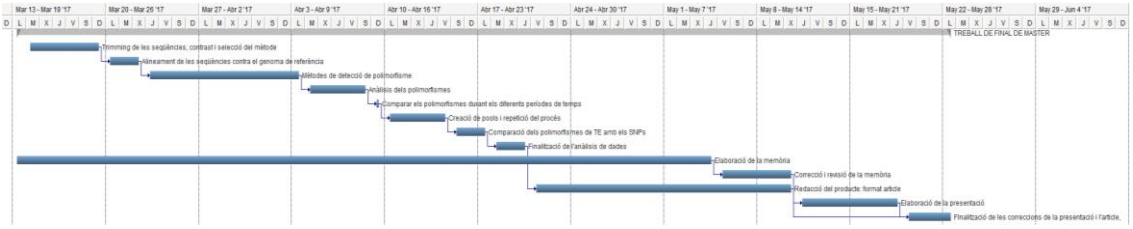

Figura 38: Planificació inicial

El principal punt modificat va ser la comparació dels polimorfismes de transposons amb *SNP*s que no s'ha pogut realitzar degut a que aquesta anàlisis realitzada per un altra grup encara no està complert i no se'n tenen els resultats. També s'ha modificat els dies destinats als mètodes de detecció de polimorfismes que es van realitzar durant el temps destinat a la comparació dels polimorfismes de Transposons. En general, la planificació s'ha complert abans del temps planificat entre altres coses al no poder realitzar la comparació, això va permetre realitzar la detecció de polimorfismes de transposons en diferents ecotips.

La metodologia seguida en el treball es considera que ha estat adequada per tal de poder arribar a la finalitat desitjada. Aconseguint realitzar les anàlisis de forma adequada. Sí però que han calgut introduir tot un seguit de canvis del plantejat inicialment per tal d'arribar a l'èxit del treball, com ha estat la introducció de l'estudi de diferents ecotips de *P. patens.*

Pel que fa a les línies de futur d'aquest treball la intenció és ampliar el anàlisis en diferents ecotips a mesura que es vagin publicant noves seqüenciacions d'aquests, el següent ecotip que s'estudiarà serà l'ecotip *Kaskaskian (Rensing et al., 2017):* el qual ja s'ha analitzat els *SNP*s però a dia d'avui no es troben les dades públiques. S'han demanat les dades per tal de realitzar l'anotació en un futur immediat.

La detecció de polimorfismes de transposons en *Villersexel* i *Reute* s'utilitzarà en el laboratori d'anàlisis i estructura de l'evolució de genomes de plantes per tal d'estudiar l'impacte dels transposons sobre els gens propers. Es seleccionaran aquells transposons polimòrfics propers als gens (< 1 kb) que puguin produir potencialment un fenotip i s'analitzaran mitjançant RT *PCR*. Un cop s'hagi analitzat es compararan les dades entre els diferents ecotips i finalment s'intentarà retirar mitjançant el sistema CRISPR-Cas9 aquests polimorfismes. Això servirà per comprovar si efectivament aquests polimorfismes alteren la expressió dels gens i si es pot recuperar la expressió prèvia retirant de forma selectiva aquests elements polimòrfics.

Finalment, encara suposa una gran dificultat dur a terme la detecció d'insercions de transposons dintre d'altres transposons. Aquí hem vist que una alternativa és córrer el programa de detecció *jitterbug* amb una anotació parcial dels elements, però tot i això no hem pogut detectar, per exemple, tots aquells elements *RLG1* polimòrfics dintre d'altres elements *RLG1*. Durant un futur, també s'intentarà buscar mètodes que permetin aquest tipus de deteccions.

6. Glossari

Elements transposables
Elements transposables o transposons són seqüències de DNA que tenen la capacitat de desplaçar-se al llarg del genoma de les cèl·lules d'un organisme, podent causar mutacions durant aquest procés.

Retrotransposons
Elements transposables de tipus I amb la capacitat de copiar-se i generar noves còpies al llarg del genoma, tenen capacitat de transcriure un RNA i retrotranscriure's i mitjançant una transcriptasa inversa.

Insercions
Presència d'una seqüència en una nova seqüenciació que no es troba en la mateixa posició en el genoma de referència.

Delecions
Absència d'una seqüència en una nova seqüenciació que sí que és present en el genoma de referència.

Polimorfismes:
Conjunt de canvis estructurals o nucleotídics en una nova seqüenciació respecte el genoma de referència.

RLG1
Retrotransposó de tipus Gypsy de la família 1 del genoma de *Physcomitrella patens.*

Zigositat
Grau de similaritat dels al·lels per una característica en un organisme

Coverage

Quantitat de *reads* respecte el genoma de referència. Normalment es té en compte el *coverage* com la quantitat de vegades que es troba representat el genoma de referència, essent per exemple un *coverage de 10x* que es troba representada 10 vegades la mida del genoma.

Reads
Seqüències curtes de DNA. Normalment són l'*output* directe de les màquines de seqüenciació.

Ecotips
Subdivisió genètica i ecològica d'una espècie.

LTR
Seqüències idèntiques de DNA que es repeteixen al llarg dels retrotransposons.

Physcomitrella patens:
Espècie briòfita (molsa) utilitzada com a organisme model en estudis d'evolució vegetal, desenvolupament i fisiologia.

7. Bibliografia

Webgrafia

1) https://www.ncbi.nlm.nih.gov/genome/browse/ [21/05/2017]
2) https://www.bioinformatics.babraham.ac.uk/projects/fastqc/Help/3%20Analysis%20Modules/12%20Per%20Tile%20Sequence%20Quality.html [03/4/2017]
3) https://github.com/relipmoc/*skewer* [03/4/2017]
4) https://phytozome.jgi.doe.gov/pz/portal.html#!info?alias=Org_Ppatens [4/05/2017]
5) http://www.moss-stock-center.org/strains.html [08/05/2017]
6) http://bioinfo.ut.ee/primer3-0.4.0/ [1/5/2017]

Bibliografia

- ADAMS, Mark D., *et al*. The genome sequence of Drosophila melanogaster. Science, 2000, 287.5461: 2185-2195.
- ALTSCHUL, Stephen F., *et al*. Basic local alignment search tool. Journal of molecular biology, 1990, 215.3: 403-410.
- ANDREWS, Simon, *et al*. FastQC: a quality control tool for high throughput sequence data. 2010.
- ARABIDOPSIS GENOME INITIATIVE, *et al*. Analysis of the genome sequence of the flowering plant Arabidopsis thaliana. nature, 2000, 408.6814: 796.
- BENNETT, Simon. Solexa ltd. *Pharmacogenomics*, 2004, 5.4: 433-438.
- BLATTNER, Frederick R., *et al*. The complete genome sequence of Escherichia coli K-12. science, 1997, 277.5331: 1453-1462.
- BOLGER, Anthony M.; LOHSE, Marc; USADEL, Bjoern. Trimmomatic: a flexible trimmer for Illumina sequence data. Bioinformatics, 2014, btu170.
- CHEN, Jinfeng, *et al*. RelocaTE2: a high resolution transposable element insertion site mapping tool for population resequencing. PeerJ, 2017, 5: e2942.
- ELLINGHAUS, David; KURTZ, Stefan; WILLHOEFT, Ute. LTRharvest, an efficient and flexible software for de novo detection of LTR retrotransposons. BMC bioinformatics, 2008, 9.1: 18.
- EWING, Adam D. Transposable element detection from whole genome sequence data. Mobile DNA, 2015, 6.1: 24.
- FESCHOTTE, Cédric; JIANG, Ning; WESSLER, Susan R. Plant transposable elements: where genetics meets genomics. Nature Reviews Genetics, 2002, 3.5: 329-341.
- GILLY, Arthur, *et al*. TE-Tracker: systematic identification of transposition events through whole-genome resequencing. BMC bioinformatics, 2014, 15.1: 377.
- GREGORY, Simon G., *et al*. A physical map of the mouse genome. Nature, 2002, 418.6899:743-750.

- HÉNAFF, Elizabeth, *et al*. Jitterbug: somatic and germline transposon insertion detection at single-nucleotide resolution. BMC genomics, 2015, 16.1: 768.
- HISS, Manuel, *et al*. Sexual reproduction, sporophyte development and molecular variation in the model moss *Physcomitrella patens*: introducing the ecotype Reute. The Plant Journal, 2017, 90.3: 606-620.
- JIANG, Hongshan, *et al*. Skewer: a fast and accurate adapter trimmer for next-generation sequencing paired-end reads. BMC bioinformatics, 2014, 15.1: 182.
- KEANE, Thomas M.; WONG, Kim; ADAMS, David J. RetroSeq: transposable element discovery from next-generation sequencing data. Bioinformatics, 2013, 29.3: 389-390.
- LI, Heng, *et al*. The sequence alignment/map format and SAMtools. Bioinformatics, 2009, 25.16: 2078-2079.
- LI, Heng. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. arXiv preprint arXiv:1303.3997, 2013.
- LI, Heng; DURBIN, Richard. Fast and accurate long-read alignment with Burrows–Wheeler transform. Bioinformatics, 2010, 26.5: 589-595.
- LISCH, Damon. How important are transposons for plant evolution?. Nature Reviews Genetics, 2013, 14.1: 49-61.
- MÜLLER, Stefanie J., *et al*. Can mosses serve as model organisms for forest research?. Annals of Forest Science, 2016, 73.1: 135-146.
- NEKRUTENKO, Anton; LI, Wen-Hsiung. Transposable elements are found in a large number of human protein-coding genes. TRENDS in Genetics, 2001, 17.11: 619-621.
- NIELSEN, Rasmus, *et al*. Genotype and SNP calling from next-generation sequencing data. Nature Reviews Genetics, 2011, 12.6: 443-451.
- OKONECHNIKOV, Konstantin, *et al*. Unipro UGENE: a unified bioinformatics toolkit. Bioinformatics, 2012, 28.8: 1166-1167.
- PORCEDDU, Andrea, *et al*. Development of S-SAP markers based on an LTR-like sequence from Medicago sativa L. Molecular Genetics and Genomics, 2002, 267.1: 107-114.
- RENSING Stefan *et al* P. patens chromosome assembly reveals extraordinary moss genome structure and 2 evolution 2017)(peer review)
- RENSING, Stefan A., *et al*. The *Physcomitrella* genome reveals evolutionary insights into the conquest of land by plants. Science, 2008, 319.5859: 64-69.
- RISHISHWAR, Lavanya; MARIÑO-RAMÍREZ, Leonardo; JORDAN, I. King. Benchmarking computational tools for polymorphic transposable element detection. Briefings in Bioinformatics, 2016, bbw072.
- SHENDURE, Jay; JI, Hanlee. Next-generation DNA sequencing. Nature biotechnology, 2008, 26.10: 1135-1145.
- SNIEGOWSKI, Paul D.; GERRISH, Philip J.; LENSKI, Richard E. Evolution of high mutation rates in experimental populations of E. coli. Nature, 1997, 387.6634: 703.

- STONEKING, Mark. Single nucleotide polymorphisms: From the evolutionary past.. Nature, 2001, 409.6822: 821-822.
- TARAILO-GRAOVAC, Maja; CHEN, Nansheng. Using RepeatMasker to identify repetitive elements in genomic sequences. Current Protocols in Bioinformatics, 2009, 4.10. 1-4.10. 14.
- TATUSOVA, Tatiana A.; MADDEN, Thomas L. BLAST 2 Sequences, a new tool for comparing protein and nucleotide sequences. FEMS microbiology letters, 1999, 174.2: 247-250.
- THORVALDSDÓTTIR, Helga; ROBINSON, James T.; MESIROV, Jill P. Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration. Briefings in bioinformatics, 2013, 14.2: 178-192.
- THUNG, Djie Tjwan, *et al*. Mobster: accurate detection of mobile element insertions in next generation sequencing data. Genome biology, 2014, 15.10: 488.
- YE, Kai, *et al*. Pindel: a pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads. Bioinformatics, 2009, 25.21: 2865-2871.
- ZHUANG, Jiali, *et al*. TEMP: a computational method for analyzing transposable element polymorphism in populations. Nucleic acids research, 2014, 42.11: 6826-6838.

8. Annexos

I.-Codi utilitzat per executar *SKEWER*

Skewer: Programa de filtratge de dades de *Next generation Sequencing* detecta i elimina les seqüències adaptadores necessàries per dur a terme el procés de seqüenciació, filtra per qualitat dels *read*s utilitzant l'algorisme desenvolupat pel grup que va crear el programa anomenat bit-masked k-difference matching algorithm (Jiang, Hongshan, *et al.*, 2014).

Codi utilitzat:

```
>skewer-0.2.2-linux-x86_64 --output $3 --compress --mean-quality 25 --min 35 --quiet $1 $2
```

On $1 i $2 corresponen a la parella de *fastq* previs a filtrar com a *input* i $3 al *output.* En aquest cas s'ajusta perquè la qualitat sigui de com a mínim 25 i la longitud dels *read*s sigui de com a mínim 35, valors inferiors no ens serviran per tal de trobar punts d'inserció de transposons polimòrfics.

II.-Codi utilitzat per executar *TRIMMOMATIC*

Trimmomatic: També és un programa de filtratge per *fastq* procedents de *NGS* permet filtrar en format paired end. Presenta l'avantatge que és molt més flexible que altres programes. Elimina els adaptadors de les seqüències, els palíndroms i aquells reads de baixa qualitat (Bolger, Anthony M. *et al.*,2014).

```
trimmomatic-0.33.jar PE -trimlog $1 -phred33 $2 $3 $4 $5 $6 $7
ILLUMINACLIP:/home/pvendrell/software/TruSeq3-PE.fa:2:30:10 LEADING:10 TRAILING:10
SLIDINGWINDOW:4:25 MINLEN:35
```

En aquest cal subministrar els adaptadors utilitzats. També cal ajustar les posicions de *Leading i Trailing* aquestes són les posicions que es retirarien en cas que hi hagi algun problema a l'inici o final de les seqüències així com que la longitud mínima ha de ser de 35 pb. En aquest cas els valors $2 i $3 corresponen als inputs és a dir els *fastq* provinents de *illumina* els valors $4 i $6 són els outputs que s'han filtrat i no han perdut la parella que seran els que utilitzarem en el posterior anàlisis. Finalment els $5 i $7 són aquells que després de filtrar han quedat desaparellats, aquests en el cas de realitzar per exemple un anàlisis de *SNPs* s'utilitzarien ja que podrien aparellar correctament, però en el cas dels transposons és imprescindible mantenir la parella per aquest fet en cas que una de les dos es filtri en els passos prèvis es descartaran ambdós *reads.*

III.-*Script* utilitzat per executar el filtratge amb *BWA Aln*

El script elaborat és el següent:

```
#!/bin/bash
#SBATCH --job-name=BAM_aln
#SBATCH --ntasks=1
#SBATCH --mem-per-cpu=4G
#SBATCH --cpus-per-task=6
#SBATCH -o /home/pvendrell/running_bam_aln_%j.out
#SBATCH -e /home/pvendrell/running_bam_aln_%j.err
#SBATCH --get-user-env=PWD
#SBATCH --partition=general
#SBATCH --nodelist=huberman
```

```
source /opt/Modules/3.2.9/init/Modules4bash.sh
module load conda
source activate BWA-0.7.15
module load samtools/0.1.18-sl61
genome=$1
READS1=$2
READS2=$3
sample=$4

BWA index $READS1
BWA index $READS2
BWA aln -t 6 -n 5 -o 1 -e 3 -f $READS1.sai $genome $READS1
BWA aln -t 6 -n 5 -o 1 -e 3 -f $READS2.sai $genome $READS2
BWA sampe $genome $READS1.sai $READS2.sai $READS1 $READS2 > $sample.r1-r2.sw.newtag.sam
samtools view -b -t $genome.fai -S -o $sample.r1-r2.sw.nsorted.newtag.aln.bam $sample.r1-
r2.sw.newtag.sam
samtools sort $sample.r1-r2.sw.nsorted.newtag.aln.bam $sample.r1-r2.sw.psorted.aln.newtag
samtools index $sample.r1-r2.sw.psorted.aln.newtag.bam

echo "## $t DONE!"
source deactivate BWA-0.7.15
```

IV.-*Script* utilitzat per executar el filtratge amb *BWA mem*

```
#!/bin/bash
#SBATCH --job-name=BAM_mem
#SBATCH --ntasks=1
#SBATCH --mem-per-cpu=4G
#SBATCH --cpus-per-task=6
#SBATCH -o /home/pvendrell/running_bam_mem_%j.out
#SBATCH -e /home/pvendrell/running_bam_mem_%j.err
#SBATCH --get-user-env=PWD
#SBATCH --partition=general
#SBATCH --nodelist=huberman
source /opt/Modules/3.2.9/init/Modules4bash.sh
module load BWA/0.7.5a
module load samtools/0.1.18-sl61

genome=$1
READS2=$3
sample=$4

BWA mem -t 6 $genome $READS1 $READS2 > $sample.r1-r2.sw.sam

samtools view -b -t $genome.fai -S -o $sample.r1-r2.sw.nsorted.bam $sample.r1-r2.sw.sam

samtools sort $sample.r1-r2.sw.nsorted.bam $sample.r1-r2.sw.psorted

samtools index $sample.r1-r2.sw.psorted.bam

echo "## $t DONE!"
```

V.- Script elaborat per executar *Pindel al servidor*

```
#!/bin/bash -x
#SBATCH --job-name=pindel
#SBATCH --ntasks=1
#SBATCH --mem-per-cpu=5G
#SBATCH --cpus-per-task=5
#SBATCH -o /scratch/074-arabidopsis-MITEs/running%j.out
#SBATCH -e /scratch/074-arabidopsis-MITEs/running%j.err
#SBATCH --get-user-env=PWD
#SBATCH --partition=fatnodes
# SBATCH --nodelist=huberman


/bin/date
uname -n
source /opt/Modules/3.2.9/init/Modules4bash.sh
module load pindel/0.2.4t
```

```
pindel -f /scratch/074-arabidopsis-
MITEs/physco/Physcomitrella_patens.main_genome.scaffolds.fasta -i /scratch/074-arabidopsis-
MITEs/physco/pindel/config_filebamcombinated2010.txt -x 5 -r false -t false -T 6 -A 35 -o
/scratch/074-arabidopsis-MITEs/physco/pindel/2010/pindel_results_2010

pindel -f /scratch/074-arabidopsis-
MITEs/physco/Physcomitrella_patens.main_genome.scaffolds.fasta -i /scratch/074-arabidopsis-
MITEs/physco/pindel/config_filebamcombinated2016_2.txt -x 5 -r false -t false -T 6 -A 35 -o
/scratch/074-arabidopsis-MITEs/physco/pindel/2016/pindel_results_2016_2

/bin/date
exit 0;
```

VI.- Script elaborat per entrecreuar l'anotació de transposons amb les delecions detectades

```
#!/bin/bash

sample=$1
TE=$2 ##TE_annotation in gff3 format



# STEP 1: Filter by size (less than 25 kb and bigger than 200 bp) and convert it to gff3
##changed 200 to 100
grep -w D $sample | awk 'BEGIN{OFS="\t"}{if($11-$10 > 100 && $11-$10 < 25000){print
$8,"TE_deletion","pindel",$10,$11,".","+",".","score="$25";total_reads_uniq="$17";fwd_reads_uniq
="$20";rev_reads_uniq="$23";length="$11-$10}}' > $sample.pindel-results.DEL.l200L25k.gff3

# STEP 2: Filter by overlapping deletions with at least one annotated TE (real number of
deletions)

intersectBed -a $sample.pindel-results.DEL.l200L25k.gff3 -b $TE -u > $sample.pindel-
results.DEL.l100L25k.TE.gff3

#### -u imprimeix unicament un cop si es detecta un overlap d'una deleció amb més d'un TE

# STEP 3: Number of overlaps between a deletion and annotated_TEs in reference genome

intersectBed -a $sample.pindel-results.DEL.l100L25k.TE.gff3 -b $TE -c > $sample.pindel-
results.DEL.l100L25k.TE-counts.gff3

# STEP 4: For each deletion, add the tag of all overlapping annotated_TEs

intersectBed -a $sample.pindel-results.DEL.l100L25k.TE.gff3 -b $TE -wao | awk 'BEGIN
{FS="\t";OFS="\t"}{print$1,$2,$3,$4,$5,$6,$7,$8,$9";"$18}'>$sample.pindel-
results.DEL.l100L25k.TE-match.gff3
```

VII-Taula de polimorfismes durant diferents períodes de temps en les mostres individuals

2010 RLG1 X COVERAGE 13.02									
chr	**polymorphis**	**program**	**start**	**end**	**.**	**sense**	**.**	**description**	**zygosity**
Chr09	jitterbug	TE_insertion	5429905	5430195	.	.	.	supporting_fwd_reads	1
Chr13	jitterbug	TE_insertion	16551158	16551602	.	.	.	supporting_fwd_reads	-1
Chr20	jitterbug	TE_insertion	6770221	6770572	.	.	.	supporting_fwd_reads	1
2010 RLG1 Y COVERAGE 10.20									
chr	**polymorphis**	**program**	**start**	**end**	**.**	**sense**	**.**	**description**	**zygosity**
Chr02	jitterbug	TE_insertion	18079264	18080034	.	.	.	supporting_fwd_reads	-1
Chr09	jitterbug	TE_insertion	15844736	15845222	.	.	.	supporting_fwd_reads	-1
Chr13	jitterbug	TE_insertion	16551097	16551634	.	.	.	supporting_fwd_reads	1
Chr20	jitterbug	TE_insertion	6770230	6770555	.	.	.	supporting_fwd_reads	1
2010 RLG1 Z COVERAGE 10.70									
chr	**polymorphis**	**program**	**start**	**end**	**.**	**sense**	**.**	**description**	**zygosity**
Chr09	jitterbug	TE_insertion	5429874	5430222	.	.	.	supporting_fwd_reads	-1
Chr14	jitterbug	TE_insertion	12145602	12146453	.	.	.	supporting_fwd_reads	1
Chr15	jitterbug	TE_insertion	7628498	7628593	.	.	.	supporting_fwd_reads	-1
2010 RLG1 0 COVERAGE 10.53									
chr	**polymorphis**	**program**	**start**	**end**	**.**	**sense**	**.**	**description**	**zygosity**
Chr03	jitterbug	TE_insertion	21809619	21810471	.	.	.	supporting_fwd_reads	1
Chr06	jitterbug	TE_insertion	13845356	13846064	.	.	.	supporting_fwd_reads	-1
Chr09	jitterbug	TE_insertion	5429835	5430198	.	.	.	supporting_fwd_reads	1
Chr13	jitterbug	TE_insertion	16551151	16551658	.	.	.	supporting_fwd_reads	1
2016 TE 1 COVERAGE 11.13									
chr	**polymorphis**	**program**	**start**	**end**	**.**	**sense**	**.**	**description**	**zygosity**
Chr13	jitterbug	TE_insertion	9821514	9822223	.	.	.	supporting_fwd_reads	1
2016 TE 2 COVERAGE 10.789									
chr	**polymorphis**	**program**	**start**	**end**	**.**	**sense**	**.**	**description**	**zygosity**
Chr13	jitterbug	TE_insertion	9821514	9822223	.	.	.	supporting_fwd_reads	1
2016 TE 3 COVERAGE 11.14									
chr	**polymorphis**	**program**	**start**	**end**	**.**	**sense**	**.**	**description**	**zygosity**
Chr04	jitterbug	TE_insertion	4639081	4639241	.	.	.	supporting_fwd_reads	0,986
Chr11	jitterbug	TE_insertion	405980	406387	.	.	.	supporting_fwd_reads	1
Chr17	jitterbug	TE_insertion	3905271	3905649	.	.	.	supporting_fwd_reads	1
2016 TE 4 COVERAGE 12.27									
chr	**polymorphis**	**program**	**start**	**end**	**.**	**sense**	**.**	**description**	**zygosity**
Chr07	jitterbug	TE_insertion	6799625	6799732	.	.	.	supporting_fwd_reads	0,946
2016 TE 6 COVERAGE 11.85									
chr	**polymorphis**	**program**	**start**	**end**	**.**	**sense**	**.**	**description**	**zygosity**
Chr07	jitterbug	TE_insertion	6799625	6799732	.	.	.	supporting_fwd_reads	0,946
2016 RLG 1 1									
chr	**polymorphis**	**program**	**start**	**end**	**.**	**sense**	**.**	**description**	**zygosity**
Chr01	jitterbug	TE_insertion	29805870	29806755	.	.	.	supporting_fwd_reads	-1
Chr02	jitterbug	TE_insertion	11396954	11397576	.	.	.	supporting_fwd_reads	1
Chr09	jitterbug	TE_insertion	15844720	15845183	.	.	.	supporting_fwd_reads	1
Chr11	jitterbug	TE_insertion	14865105	14865214	.	.	.	supporting_fwd_reads	-1
Chr13	jitterbug	TE_insertion	16551130	16551616	.	.	.	supporting_fwd_reads	1
Chr19	jitterbug	TE_insertion	5743712	5744092	.	.	.	supporting_fwd_reads	-1
Chr21	jitterbug	TE_insertion	12851302	12851732	.	.	.	supporting_fwd_reads	1
Chr22	jitterbug	TE_insertion	7146789	7147106	.	.	.	supporting_fwd_reads	-1
Chr24	jitterbug	TE_insertion	7877282	7877778	.	.	.	supporting_fwd_reads	-1
2016 RLG 1 9									
chr	**polymorphis**	**program**	**start**	**end**	**.**	**sense**	**.**	**description**	**zygosity**
Chr21	jitterbug	TE_insertion	3473314	3473616	.	.	.	supporting_fwd_reads	1
2016 RLG 1 8									
chr	**polymorphis**	**program**	**start**	**end**	**.**	**sense**	**.**	**description**	**zygosity**
Chr06	jitterbug	TE_insertion	10030240	10030657	.	.	.	supporting_fwd_reads	1
Chr20	jitterbug	TE_insertion	6770206	6770542	.	.	.	supporting_fwd_reads	-1
Chr21	jitterbug	TE_insertion	12851451	12851932	.	.	.	supporting_fwd_reads	1
2016 RLG 1 8									
chr	**polymorphis**	**program**	**start**	**end**	**.**	**sense**	**.**	**description**	**zygosity**
Chr06	jitterbug	TE_insertion	10030236	10030584	.	.	.	supporting_fwd_reads	0,75
2016 RLG 1 1									
chr	**polymorphis**	**program**	**start**	**end**	**.**	**sense**	**.**	**description**	**zygosity**
Chr05	jitterbug	TE_insertion	8132609	8132952	.	.	.	supporting_fwd_reads	-1
Chr07	jitterbug	TE_insertion	6516325	6516513	.	.	.	supporting_fwd_reads	-1
Chr07	jitterbug	TE_insertion	15486271	15487002	.	.	.	supporting_fwd_reads	-1
Chr09	jitterbug	TE_insertion	5429901	5430195	.	.	.	supporting_fwd_reads	-1
Chr12	jitterbug	TE_insertion	8175965	8176082	.	.	.	supporting_fwd_reads	-1
Chr15	jitterbug	TE_insertion	7628501	7628605	.	.	.	supporting_fwd_reads	-1
Chr18	jitterbug	TE_insertion	2841249	2841806	.	.	.	supporting_fwd_reads	-1
Chr19	jitterbug	TE_insertion	3451873	3452159	.	.	.	supporting_fwd_reads	-1
Chr19	jitterbug	TE_insertion	9536373	9536950	.	.	.	supporting_fwd_reads	-1
Chr19	jitterbug	TE_insertion	12419014	12419576	.	.	.	supporting_fwd_reads	1
Chr19	jitterbug	TE_insertion	15152615	15152895	.	.	.	supporting_fwd_reads	1
2016 RLG1 3									
chr	**polymorphis**	**program**	**start**	**end**	**.**	**sense**	**.**	**description**	**zygosity**
Chr09	jitterbug	TE_insertion	5429858	5430177	.	.	.	supporting_fwd_reads	-1
2016 RLG1 4									
chr	**polymorphis**	**program**	**start**	**end**	**.**	**sense**	**.**	**description**	**zygosity**
Chr02	jitterbug	TE_insertion	18079345	18079859	.	.	.	supporting_fwd_reads	1
Chr13	jitterbug	TE_insertion	16551128	16551623	.	.	.	supporting_fwd_reads	-1
2016 RLG1 6									
chr	**polymorphis**	**program**	**start**	**end**	**.**	**sense**	**.**	**description**	**zygosity**
Chr13	jitterbug	TE_insertion	16551148	16551602	.	.	.	supporting_fwd_reads	-1
Chr20	jitterbug	TE_insertion	6770242	6770530	.	.	.	supporting_fwd_reads	1

VIII.- Oligonucleòtids dissenyats per detectar polimorfismes d'elements *RLG1*

NAME	INF	SEQUENCE	DESCRIPTION	Tm1	Length	GRANSDEN/VILLERSEXEL	%G/C	Cross dimer (kcal/mol)	Self dimer (Kcal/mol)
0PV3	polimorphism RLG1 chr 3 fw	TCAAACAATGGCAAAGGATG	Check for RLG1 insertion in Villersexel or absence in gransden in Chr 3 / 900 pb	56,76	20	900pb	40	-4,43	/
oPV4	polimorphism RLG1 chr 3 rev	GCACCCAAGGCATACAAAAT	Check for RLG1 insertion in Villersexel or absence in gransden in Chr 3 /900 pb	57,67	20	900pb	45	-4,43	/
oPV5	polimorphism RLG1 chr 7,1 fw	GACAAGATGTCCTGAATGCAAA	Check for RLG1 deletion in Villersexel or absence in gransden in Chr 7 /9,5 kb	57,72	22	9,5kb	41	-8	-8,05
oPV6	polimorphism RLG1 chr 7,1 rev	CAATCCTTGCCACTTTTGTC	Check for RLG1 deletion in Villersexel or absence in gransden in Chr 7 /9,5 kb	55,52	20	9,5kb	45	-8	-4,9
oPV7	polimorphism RLG1 chr 7,2 fw	TGTGGTTTAATTTTGGACCTTTT	Check for RLG1 insertion in Villersexel or absence in gransden in Chr 7 / 446 pb	58,05	23	446 pb	30	/	-5,36
oPV8	polimorphism RLG1 chr 7,2 rev	GGGAATGTTTTGCTCATGCT	Check for RLG1 insertion in Villersexel or absence in gransden in Chr 7/ 446 pb	57,58	20	446 pb	45	/	-6,38
oPV9	polimorphism RLG1 chr 8 fw and rev	CAGCCTCTCGTTCTTGTTCT	Check for RLG1 deletion in Villersexel or absence in gransden in Chr 8/ 10 kb	54,42	20	10kb	50	/	/
oPV10	polimorphism RLG1 chr 11 fw	CAAAAACAAAACCAAAACCAA	Check for RLG1 deletion in Villersexel or absence in gransden in Chr 11/ 1kb	56,02	21	1kb	28	-4,9	/
oPV11	polimorphism RLG1 chr 11 rev	ACGTTGCGATCACAGAAATG	Check for RLG1 insertion in Villersexel or absence in gransden in Chr 11 / 1kb	56,6	20	1kb	45	-4,9	-6,3
oPV12	polimorphism RLG1 chr 18 fw	AAAAACTCATCGGTCAAAAATCA	Check for RLG1 deletion in Villersexel or absence in gransden in Chr 18/length 16 kb	58,6	23	16kb	30	-4,53	/
oPV13	polimorphism RLG1 chr 18 rev	TGAAGCAAAAATTGGTGAAAAA	Check for RLG1 deletion in Villersexel or absence in gransden in Chr 18/ length 16 kb	58,2	22	16kb	27	-4,53	/

Les condicions de les *PCR* eren les següents:

Cicle	Temperatura		Temps	Descripció	Repeticions
1	95ºC		5 minuts	Desnaturalització del DNA	1
2	95ºC		45 segons	Desnaturalització del DNA	35
3	56ºC		30 segons	*Annealing del* DNA	
4	65ºC		15 minuts	Amplificació de la polimerasa *Long Amp*	
5	65ºC		20 minuts	Amplificació de la polimerasa *Long Amp*	1
6	12ºC		Perpetu	Finalització de la amplificació i conservació en fred per evitar la degradació del ADN i evitar amplificacions no previstes	1

IX.-Taula dels polimorfismes dels elements *RLG1* en els diferents períodes de temps

2010 jitterbug RLG1											
											ZIGOSITY
Chr02	jitterbug	TE	insertion	2732725	2733318	.	.	.	supporting	fv	0,111
Chr02	jitterbug	TE	insertion	11396979	11397546	.	.	.	supporting	fv	1
Chr02	jitterbug	TE	insertion	18079340	18079826	.	.	.	supporting	fv	1
Chr04	jitterbug	TE	insertion	5127646	5127982	.	.	.	supporting	fv	-1
Chr04	jitterbug	TE	insertion	9579073	9579986	.	.	.	supporting	fv	1
Chr05	jitterbug	TE	insertion	6104984	6105216	.	.	.	supporting	fv	0,133
Chr05	jitterbug	TE	insertion	18644160	18644259	.	.	.	supporting	fv	-1
Chr06	jitterbug	TE	insertion	10030245	10030568	.	.	.	supporting	fv	0,6
Chr06	jitterbug	TE	insertion	13845451	13845964	.	.	.	supporting	fv	1
Chr07	jitterbug	TE	insertion	9146199	9146337	.	.	.	supporting	fv	-1
Chr07	jitterbug	TE	insertion	9720128	9720749	.	.	.	supporting	fv	1
Chr09	jitterbug	TE	insertion	15844736	15845222	.	.	.	supporting	fv	1
Chr11	jitterbug	TE	insertion	1844123	1844560	.	.	.	supporting	fv	-1
Chr12	jitterbug	TE	insertion	38662	39121	.	.	.	supporting	fv	0,125
Chr12	jitterbug	TE	insertion	16063071	16063703	.	.	.	supporting	fv	1
Chr13	jitterbug	TE	insertion	9821451	9821954	.	.	.	supporting	fv	1
Chr13	jitterbug	TE	insertion	16551158	16551602	.	.	.	supporting	fv	1
Chr14	jitterbug	TE	insertion	12145602	12146453	.	.	.	supporting	fv	1
Chr15	jitterbug	TE	insertion	7628498	7628593	.	.	.	supporting	fv	-1
Chr16	jitterbug	TE	insertion	2307135	2307412	.	.	.	supporting	fv	0,286
Chr17	jitterbug	TE	insertion	3905296	3905705	.	.	.	supporting	fv	1
Chr17	jitterbug	TE	insertion	15342811	15343318	.	.	.	supporting	fv	1
Chr18	jitterbug	TE	insertion	12787798	12788554	.	.	.	supporting	fv	-1
Chr19	jitterbug	TE	insertion	3465554	3466210	.	.	.	supporting	fv	0,045
Chr19	jitterbug	TE	insertion	12142371	12142765	.	.	.	supporting	fv	-1
Chr19	jitterbug	TE	insertion	15152628	15152833	.	.	.	supporting	fv	1
Chr21	jitterbug	TE	insertion	12851380	12851736	.	.	.	supporting	fv	1
Chr22	jitterbug	TE	insertion	3153344	3153588	.	.	.	supporting	fv	-1
Chr24	jitterbug	TE	insertion	13015738	13016136	.	.	.	supporting	fv	1
Chr26	jitterbug	TE	insertion	2032912	2033355	.	.	.	supporting	fv	-1
	30 insertions	15 zygosity=1									
				2016_1 jitterbug RLG1							
											ZIGOSITY
Chr01	jitterbug	TE	insertion	27981775	27982165	.	.	.	supporting	fv	0,059
Chr02	jitterbug	TE	insertion	11396954	11397519	.	.	.	supporting	fv	1
Chr02	jitterbug	TE	insertion	18079366	18079824	.	.	.	supporting	fv	1
Chr03	jitterbug	TE	insertion	21201854	21202111	.	.	.	supporting	fv	0,429
Chr03	jitterbug	TE	insertion	22871185	22871681	.	.	.	supporting	fv	0,5
Chr05	jitterbug	TE	insertion	6104986	6105212	.	.	.	supporting	fv	0,214
Chr06	jitterbug	TE	insertion	13845267	13845871	.	.	.	supporting	fv	1
Chr09	jitterbug	TE	insertion	5429903	5430142	.	.	.	supporting	fv	1
Chr09	jitterbug	TE	insertion	15844720	15845152	.	.	.	supporting	fv	1
Chr11	jitterbug	TE	insertion	9536662	9537095	.	.	.	supporting	fv	1
Chr11	jitterbug	TE	insertion	14865106	14865208	.	.	.	supporting	fv	-1
Chr12	jitterbug	TE	insertion	4230158	4230818	.	.	.	supporting	fv	0,091
Chr12	jitterbug	TE	insertion	16063061	16063609	.	.	.	supporting	fv	1
Chr13	jitterbug	TE	insertion	9821480	9821966	.	.	.	supporting	fv	1
Chr13	jitterbug	TE	insertion	16551130	16551616	.	.	.	supporting	fv	1
Chr15	jitterbug	TE	insertion	2185152	2185607	.	.	.	supporting	fv	1
Chr15	jitterbug	TE	insertion	10630018	10630850	.	.	.	supporting	fv	1
Chr16	jitterbug	TE	insertion	2306901	2307314	.	.	.	supporting	fv	0,174
Chr17	jitterbug	TE	insertion	3905191	3905704	.	.	.	supporting	fv	1
Chr17	jitterbug	TE	insertion	15342794	15343323	.	.	.	supporting	fv	1
Chr19	jitterbug	TE	insertion	5743712	5744092	.	.	.	supporting	fv	-1
Chr19	jitterbug	TE	insertion	12419070	12419629	.	.	.	supporting	fv	1
Chr19	jitterbug	TE	insertion	15152580	15152829	.	.	.	supporting	fv	1
Chr20	jitterbug	TE	insertion	6770236	6770522	.	.	.	supporting	fv	1
Chr21	jitterbug	TE	insertion	11338116	11338593	.	.	.	supporting	fv	1
Chr24	jitterbug	TE	insertion	12851451	12851732	.	.	.	supporting	fv	1
Chr22	jitterbug	TE	insertion	7146789	7147106	.	.	.	supporting	fv	-1
Chr24	jitterbug	TE	insertion	1972456	1972847	.	.	.	supporting	fv	1
Chr24	jitterbug	TE	insertion	7877282	7877778	.	.	.	supporting	fv	-1
	29 insertions	19 zygosity=1									
2016_2 jitterbug RLG1											
											ZIGOSITY
Chr02	jitterbug	TE	insertion	11397000	11397515	.	.	.	supporting	fv	1
Chr03	jitterbug	TE	insertion	21201769	21202269	.	.	.	supporting	fv	0,333
Chr04	jitterbug	TE	insertion	19981114	19981479	.	.	.	supporting	fv	0,001
Chr05	jitterbug	TE	insertion	6104953	6105218	.	.	.	supporting	fv	0,429
Chr05	jitterbug	TE	insertion	8132609	8132952	.	.	.	supporting	fv	-1
Chr06	jitterbug	TE	insertion	10030324	10030625	.	.	.	supporting	fv	0,667
Chr06	jitterbug	TE	insertion	13845320	13845791	.	.	.	supporting	fv	1
Chr07	jitterbug	TE	insertion	6516325	6516513	.	.	.	supporting	fv	-1
Chr09	jitterbug	TE	insertion	5429901	5430146	.	.	.	supporting	fv	-1
Chr09	jitterbug	TE	insertion	15844723	15845155	.	.	.	supporting	fv	1
Chr10	jitterbug	TE	insertion	855857	856025	.	.	.	supporting	fv	0,4
Chr10	jitterbug	TE	insertion	7267508	7267736	.	.	.	supporting	fv	-1
Chr10	jitterbug	TE	insertion	13070522	13070862	.	.	.	supporting	fv	-1
Chr11	jitterbug	TE	insertion	1844069	1844625	.	.	.	supporting	fv	-1
Chr11	jitterbug	TE	insertion	9536674	9536919	.	.	.	supporting	fv	0,44
Chr12	jitterbug	TE	insertion	8175965	8176082	.	.	.	supporting	fv	-1
Chr12	jitterbug	TE	insertion	16063061	16063608	.	.	.	supporting	fv	1
Chr13	jitterbug	TE	insertion	9821588	9821965	.	.	.	supporting	fv	1
Chr13	jitterbug	TE	insertion	14880495	14880920	.	.	.	supporting	fv	1
Chr13	jitterbug	TE	insertion	16551164	16551602	.	.	.	supporting	fv	1
Chr13	jitterbug	TE	insertion	17108313	17108832	.	.	.	supporting	fv	-1
Chr15	jitterbug	TE	insertion	4922170	4922795	.	.	.	supporting	fv	-1
Chr16	jitterbug	TE	insertion	15183808	15184145	.	.	.	supporting	fv	0,059
Chr17	jitterbug	TE	insertion	3905271	3905592	.	.	.	supporting	fv	1
Chr17	jitterbug	TE	insertion	5158105	5158545	.	.	.	supporting	fv	-1
Chr17	jitterbug	TE	insertion	15342783	15343388	.	.	.	supporting	fv	1
Chr18	jitterbug	TE	insertion	2841249	2841776	.	.	.	supporting	fv	-1
Chr19	jitterbug	TE	insertion	3451873	3452159	.	.	.	supporting	fv	-1
Chr19	jitterbug	TE	insertion	9536373	9536650	.	.	.	supporting	fv	-1
Chr19	jitterbug	TE	insertion	12419174	12419576	.	.	.	supporting	fv	1
Chr19	jitterbug	TE	insertion	13432192	13432553	.	.	.	supporting	fv	1
Chr19	jitterbug	TE	insertion	13793278	13793393	.	.	.	supporting	fv	0,05
Chr19	jitterbug	TE	insertion	15152615	15152895	.	.	.	supporting	fv	1
Chr20	jitterbug	TE	insertion	6770242	6770521	.	.	.	supporting	fv	1
Chr24	jitterbug	TE	insertion	11338106	11338621	.	.	.	supporting	fv	1
Chr25	jitterbug	TE	insertion	9906068	9906480	.	.	.	supporting	fv	1
	36 insertions	16 zygosity=1									
6 jitterbug RLG1											
											ZIGOSITY
Chr01	jitterbug	TE	insertion	4029513	4029816	.	.	.	supporting	fv	0,027
Chr01	jitterbug	TE	insertion	4108367	4108760	.	.	.	supporting	fv	0,023
Chr01	jitterbug	TE	insertion	4655357	4655620	.	.	.	supporting	fv	-1
Chr01	jitterbug	TE	insertion	5271553	5271948	.	.	.	supporting	fv	-1
Chr01	jitterbug	TE	insertion	6460481	6460723	.	.	.	supporting	fv	-1
Chr01	jitterbug	TE	insertion	14691075	14691276	.	.	.	supporting	fv	-1
Chr01	jitterbug	TE	insertion	18879501	18880045	.	.	.	supporting	fv	1
Chr01	jitterbug	TE	insertion	23039377	23039604	.	.	.	supporting	fv	-1
Chr01	jitterbug	TE	insertion	23960974	23961305	.	.	.	supporting	fv	-1
Chr01	jitterbug	TE	insertion	27577815	27578258	.	.	.	supporting	fv	0,038
Chr01	jitterbug	TE	insertion	27981775	27982165	.	.	.	supporting	fv	0,032
Chr02	jitterbug	TE	insertion	11397006	11397515	.	.	.	supporting	fv	1
Chr02	jitterbug	TE	insertion	12284942	12285167	.	.	.	supporting	fv	0,023
Chr02	jitterbug	TE	insertion	13227517	13227905	.	.	.	supporting	fv	1
Chr03	jitterbug	TE	insertion	21201854	21202269	.	.	.	supporting	fv	0,385
Chr03	jitterbug	TE	insertion	22871185	22871681	.	.	.	supporting	fv	0,5
Chr04	jitterbug	TE	insertion	1664607	1664922	.	.	.	supporting	fv	-1
Chr04	jitterbug	TE	insertion	3057615	3058036	.	.	.	supporting	fv	0,125
Chr05	jitterbug	TE	insertion	4307655	4307910	.	.	.	supporting	fv	0,029
Chr05	jitterbug	TE	insertion	8132609	8132952	.	.	.	supporting	fv	-1
Chr06	jitterbug	TE	insertion	11351837	11352238	.	.	.	supporting	fv	-1
Chr06	jitterbug	TE	insertion	13845320	13845791	.	.	.	supporting	fv	1
Chr07	jitterbug	TE	insertion	2080336	2080517	.	.	.	supporting	fv	0,024
Chr07	jitterbug	TE	insertion	6516325	6516513	.	.	.	supporting	fv	-1
Chr07	jitterbug	TE	insertion	7195472	7196087	.	.	.	supporting	fv	-1
Chr07	jitterbug	TE	insertion	12188359	12188721	.	.	.	supporting	fv	-1
Chr08	jitterbug	TE	insertion	1638371	1638631	.	.	.	supporting	fv	0,038
Chr08	jitterbug	TE	insertion	14992423	14992839	.	.	.	supporting	fv	-1
Chr08	jitterbug	TE	insertion	17287313	17287984	.	.	.	supporting	fv	0,026
Chr08	jitterbug	TE	insertion	17791265	17791818	.	.	.	supporting	fv	-1
Chr09	jitterbug	TE	insertion	5429903	5430142	.	.	.	supporting	fv	1
Chr09	jitterbug	TE	insertion	6014922	6015168	.	.	.	supporting	fv	0,03
Chr09	jitterbug	TE	insertion	9781050	9781347	.	.	.	supporting	fv	-1
Chr09	jitterbug	TE	insertion	11557545	11557705	.	.	.	supporting	fv	-1
Chr09	jitterbug	TE	insertion	13373053	13373842	.	.	.	supporting	fv	-1
Chr09	jitterbug	TE	insertion	15844723	15845152	.	.	.	supporting	fv	1
Chr10	jitterbug	TE	insertion	855857	856025	.	.	.	supporting	fv	0,314
Chr10	jitterbug	TE	insertion	4708822	4707257	.	.	.	supporting	fv	-1
Chr10	jitterbug	TE	insertion	12578250	12578360	.	.	.	supporting	fv	0,018
Chr10	jitterbug	TE	insertion	14244341	14244611	.	.	.	supporting	fv	-1
Chr11	jitterbug	TE	insertion	9536674	9536919	.	.	.	supporting	fv	1
Chr11	jitterbug	TE	insertion	14865106	14865208	.	.	.	supporting	fv	-1
Chr12	jitterbug	TE	insertion	4230158	4230818	.	.	.	supporting	fv	0,03
Chr12	jitterbug	TE	insertion	6753194	6753596	.	.	.	supporting	fv	0,179
Chr12	jitterbug	TE	insertion	8175965	8176082	.	.	.	supporting	fv	-1
Chr12	jitterbug	TE	insertion	8977628	8977921	.	.	.	supporting	fv	-1
Chr12	jitterbug	TE	insertion	11715351	11715897	.	.	.	supporting	fv	-1
Chr12	jitterbug	TE	insertion	14316297	14316840	.	.	.	supporting	fv	0,019
Chr12	jitterbug	TE	insertion	16063061	16063609	.	.	.	supporting	fv	1
Chr13	jitterbug	TE	insertion	9821588	9821965	.	.	.	supporting	fv	1
Chr13	jitterbug	TE	insertion	14880495	14880899	.	.	.	supporting	fv	1
Chr13	jitterbug	TE	insertion	15845267	15845569	.	.	.	supporting	fv	0,031
Chr13	jitterbug	TE	insertion	16551164	16551802	.	.	.	supporting	fv	1
Chr13	jitterbug	TE	insertion	17048077	17048513	.	.	.	supporting	fv	0,024
Chr14	jitterbug	TE	insertion	1263265	1263436	.	.	.	supporting	fv	-1
Chr14	jitterbug	TE	insertion	3657003	3657538	.	.	.	supporting	fv	0,167
Chr14	jitterbug	TE	insertion	5840490	5841328	.	.	.	supporting	fv	1
Chr14	jitterbug	TE	insertion	13650837	13651150	.	.	.	supporting	fv	1
Chr15	jitterbug	TE	insertion	2185152	2185587	.	.	.	supporting	fv	1
Chr15	jitterbug	TE	insertion	2318885	2319039	.	.	.	supporting	fv	-1
Chr15	jitterbug	TE	insertion	6034503	6034958	.	.	.	supporting	fv	0,021
Chr15	jitterbug	TE	insertion	7685765	7686357	.	.	.	supporting	fv	-1
Chr15	jitterbug	TE	insertion	10630018	10630564	.	.	.	supporting	fv	1
Chr15	jitterbug	TE	insertion	15786908	15787818	.	.	.	supporting	fv	0,083
Chr16	jitterbug	TE	insertion	5197951	5198163	.	.	.	supporting	fv	-1
Chr16	jitterbug	TE	insertion	5807735	5808391	.	.	.	supporting	fv	0,038
Chr16	jitterbug	TE	insertion	9820413	9820732	.	.	.	supporting	fv	-1
Chr16	jitterbug	TE	insertion	14365104	14365216	.	.	.	supporting	fv	0,034
Chr16	jitterbug	TE	insertion	15183808	15184145	.	.	.	supporting	fv	0,033
Chr16	jitterbug	TE	insertion	15588006	15588101	.	.	.	supporting	fv	-1
Chr17	jitterbug	TE	insertion	3905271	3905592	.	.	.	supporting	fv	1
Chr17	jitterbug	TE	insertion	4463425	4463555	.	.	.	supporting	fv	-1
Chr17	jitterbug	TE	insertion	5158105	5158545	.	.	.	supporting	fv	0,023
Chr17	jitterbug	TE	insertion	15342794	15343323	.	.	.	supporting	fv	1
Chr18	jitterbug	TE	insertion	2841249	2841776	.	.	.	supporting	fv	-1
Chr18	jitterbug	TE	insertion	6234269	6234752	.	.	.	supporting	fv	-1
Chr18	jitterbug	TE	insertion	10139585	10140086	.	.	.	supporting	fv	1
Chr18	jitterbug	TE	insertion	12213267	12213946	.	.	.	supporting	fv	1
Chr19	jitterbug	TE	insertion	2611940	2612034	.	.	.	supporting	fv	-1
Chr19	jitterbug	TE	insertion	3451873	3452159	.	.	.	supporting	fv	-1
Chr19	jitterbug	TE	insertion	4893748	4894422	.	.	.	supporting	fv	-1
Chr19	jitterbug	TE	insertion	5154045	5154215	.	.	.	supporting	fv	-1
Chr19	jitterbug	TE	insertion	5743743	5744092	.	.	.	supporting	fv	0,03
Chr19	jitterbug	TE	insertion	6958145	6958359	.	.	.	supporting	fv	-1
Chr19	jitterbug	TE	insertion	9536373	9536950	.	.	.	supporting	fv	-1
Chr19	jitterbug	TE	insertion	11486624	11481276	.	.	.	supporting	fv	0,042
Chr19	jitterbug	TE	insertion	12419174	12419576	.	.	.	supporting	fv	1
Chr19	jitterbug	TE	insertion	13432192	13432553	.	.	.	supporting	fv	1
Chr19	jitterbug	TE	insertion	13793278	13793393	.	.	.	supporting	fv	0,034
Chr19	jitterbug	TE	insertion	15152615	15152829	.	.	.	supporting	fv	1
Chr20	jitterbug	TE	insertion	2665227	2665543	.	.	.	supporting	fv	-1
Chr20	jitterbug	TE	insertion	6770242	6770521	.	.	.	supporting	fv	1
Chr20	jitterbug	TE	insertion	13344004	13344134	.	.	.	supporting	fv	-1
Chr21	jitterbug	TE	insertion	8644889	8645286	.	.	.	supporting	fv	-1
Chr21	jitterbug	TE	insertion	11382045	11382347	.	.	.	supporting	fv	-1
Chr22	jitterbug	TE	insertion	200616	201071	.	.	.	supporting	fv	-1
Chr22	jitterbug	TE	insertion	3156379	3156821	.	.	.	supporting	fv	0,026
Chr22	jitterbug	TE	insertion	3160278	3160617	.	.	.	supporting	fv	-1
Chr22	jitterbug	TE	insertion	7146789	7147106	.	.	.	supporting	fv	-1
Chr22	jitterbug	TE	insertion	10986210	10986737	.	.	.	supporting	fv	1
Chr22	jitterbug	TE	insertion	11752591	11752736	.	.	.	supporting	fv	-1
Chr23	jitterbug	TE	insertion	4506582	4506889	.	.	.	supporting	fv	0,025
Chr23	jitterbug	TE	insertion	5150242	5150896	.	.	.	supporting	fv	0,029
Chr24	jitterbug	TE	insertion	1972456	1972702	.	.	.	supporting	fv	1
Chr24	jitterbug	TE	insertion	10709381	10709778	.	.	.	supporting	fv	0,027
Chr25	jitterbug	TE	insertion	5324297	5324418	.	.	.	supporting	fv	-1
Chr25	jitterbug	TE	insertion	9801359	9801544	.	.	.	supporting	fv	-1

Annex X: Verificació dels artefactes 2010-2016

Es va fer el mateix per una altra artefacte, que està present en les mostres del 2010 i 2016, concretament el següent :

Chr13	jitterbug	TE_insertion	9821564	9822223

El procés va ser exactament el mateix que en el cas anterior, obtenint els següents oligonucleòtids:

5':ATAAAAATATGCCCATTTCTTCCTT
3':TGGATGCCATTGTGTAGTCTTATTA

S'esperava un producte de 1,4 Kb. Es van encarregar els *primers* i es va realitzar la *PCR* de validació obtenint el següent gel d'electroforesis:

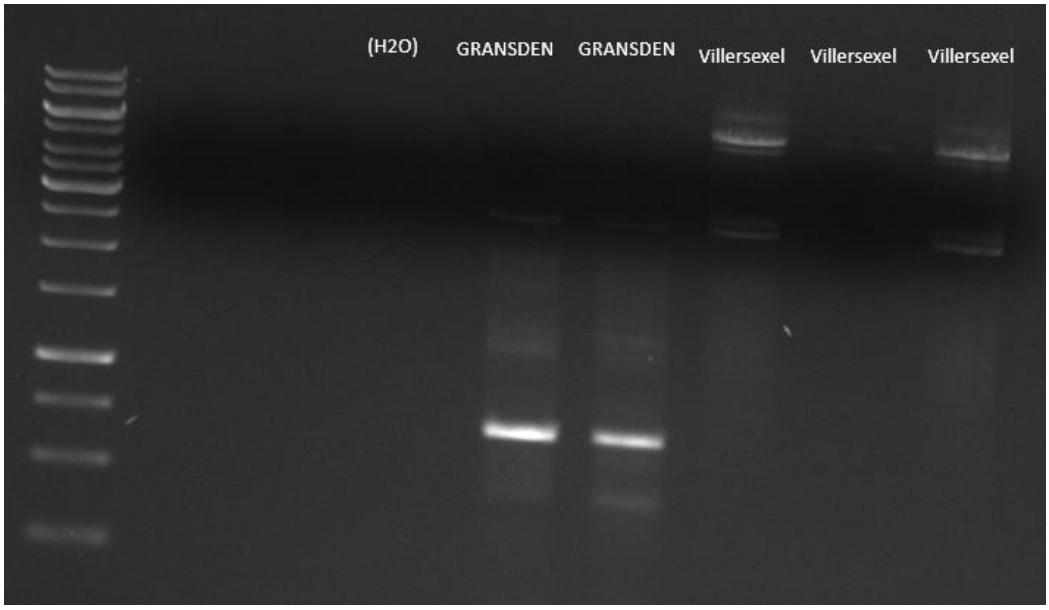

Figura 37: Patró de bandes obtinguts de la comprovació del segon artefacte

En aquest cas però no va quedar en cap cas la zona ben establerta obtenint un patró múltiple de bandes. Això pot ser degut a que es troba en una zona altament repetitiva i hagi ampliat múltiples fragments de diferents transposons per això tenim un patró de bandes tant dispars.