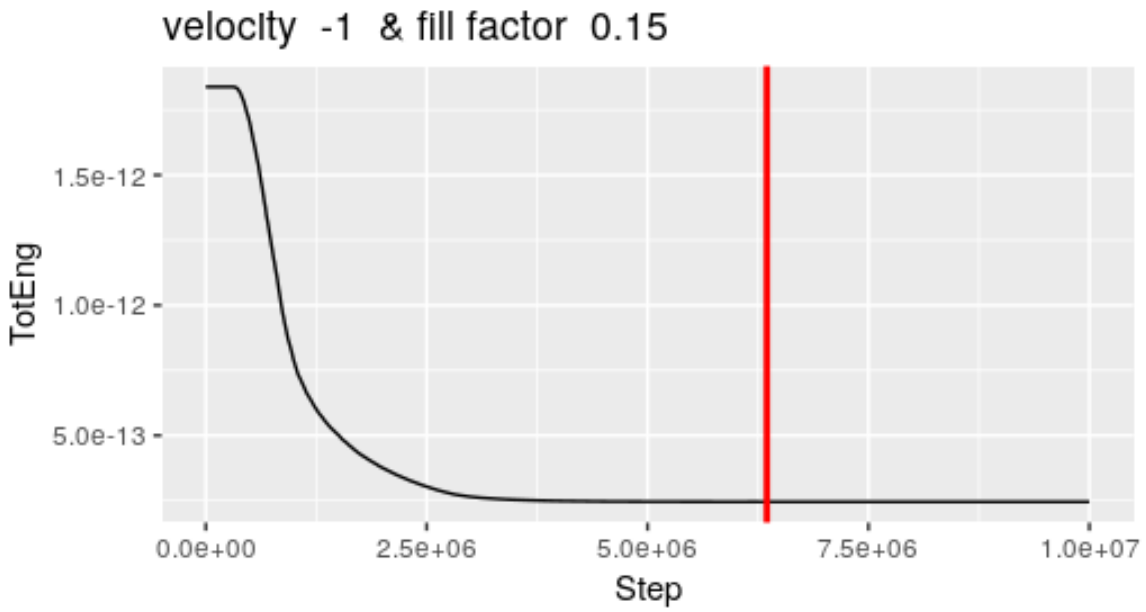
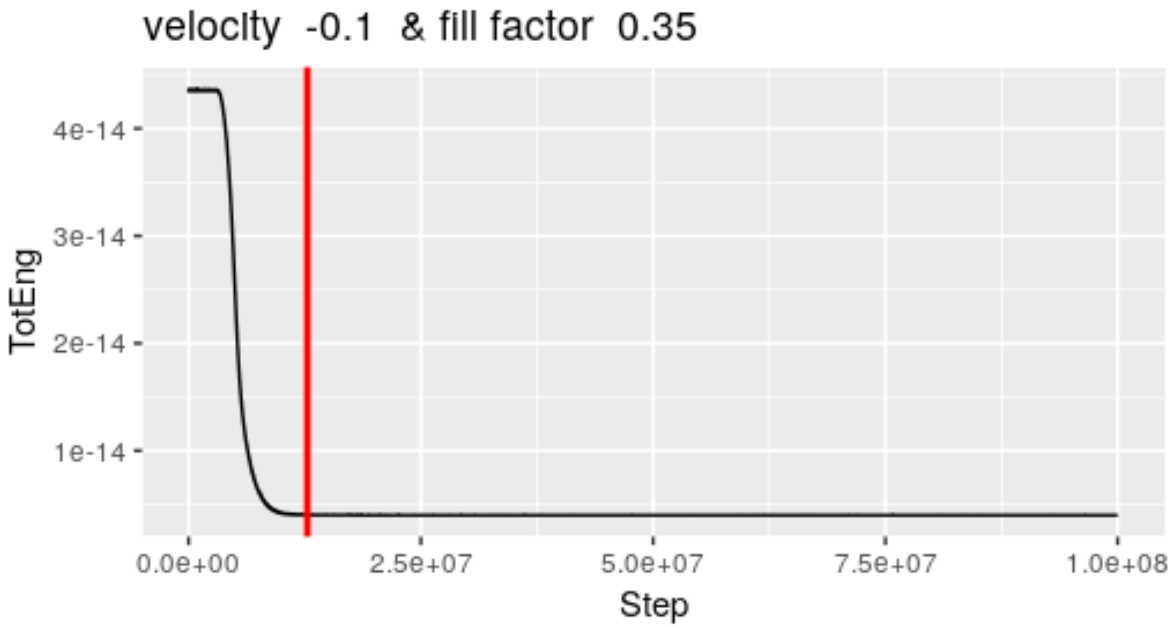
Archivos:
Log.lammps: Step Temp E_pair E_mol TotEng Press Volume (uno por cada archivo)
output*.gz: id type x y z vx vy vz omegax omegay omegaz

“script_corte_analistaR”
-Puse todos los log.lammps de los archivos en un solo data frame, clasificado por velocidad y ff. Con eso armé una función “fun1” con argumentos df y col, donde df es el df de una vel y ff del log lammps y col es el número de col que tiene la energía total. Esta función toma los 100 pasos anteriores y posteriores de un step y calcula el promedio, el cambio porcentual respecto al step actual y compara los cambios a priori y a posteriori, devolviendo el cambio porcentual en un df. La función“plot_end” con argumentos vel y ff. La función devuelve el step de corte en donde el cambio porcentual de la energía total dada por “fun1” es menor a 0.001%. Se puede pedir que devuelva el step mínimo donde esto ocurre, o utilizando la mediana, el del medio.

```
> plot_end(-1,0.15)
[1] "el step de corte sugerido es 6350000"
```

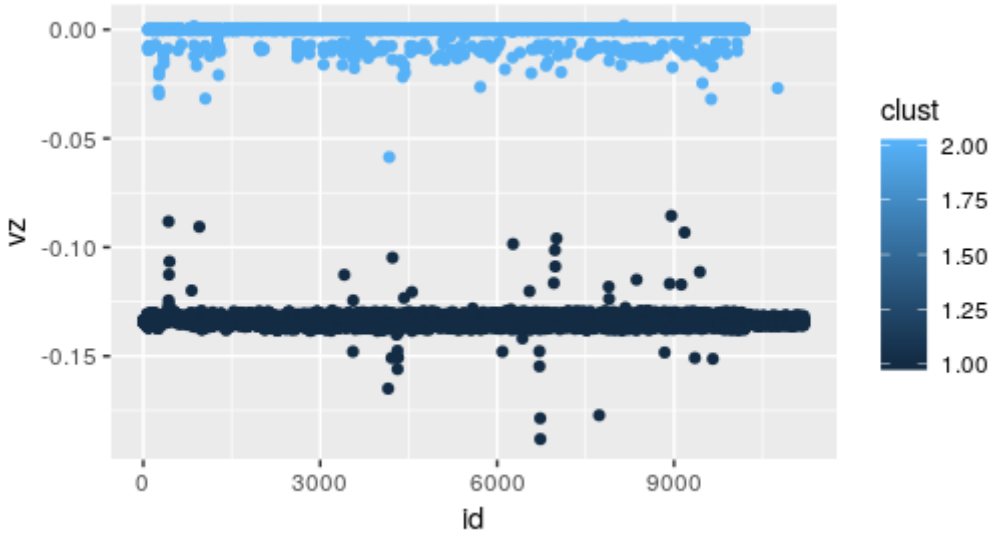



```
> plot_end(-0.1,0.35)
[1] "el step de corte sugerido es 12780000"
```



- Otra función “class_kmeans” tiene argumentos de vel, ff, y min, que es el step mínimo que se obtuvo en “plot_end”. Esta función divide los clusters del output sugerido como corte utilizando kmeans y devuelve el gráfico de separación.

```
> class_kmeans(-1, 0.15, 6350000)
[1] "DIRECTORY:"
[1] "/home/daniela/simulaciones_daniela/granular_small_0.15_-1"
```

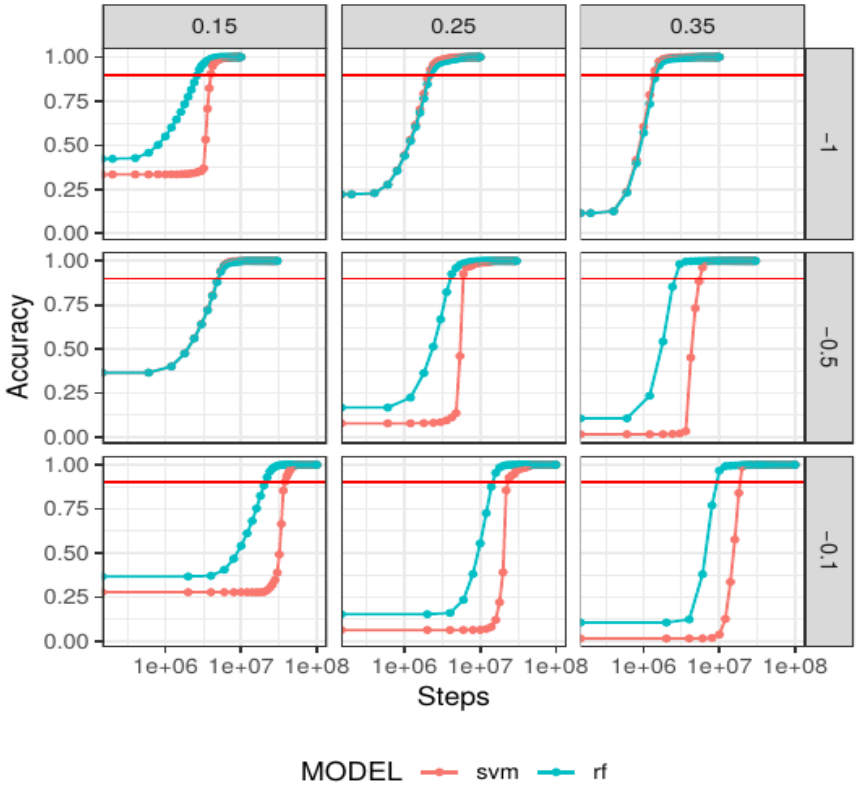


“Prueba2”
-Usando la función “plot_end” modificada, armé una matriz que tenga ff, vel y steps de cortes para todas las simulaciones.
-Usando “class_kmeans” armé un multiplot.

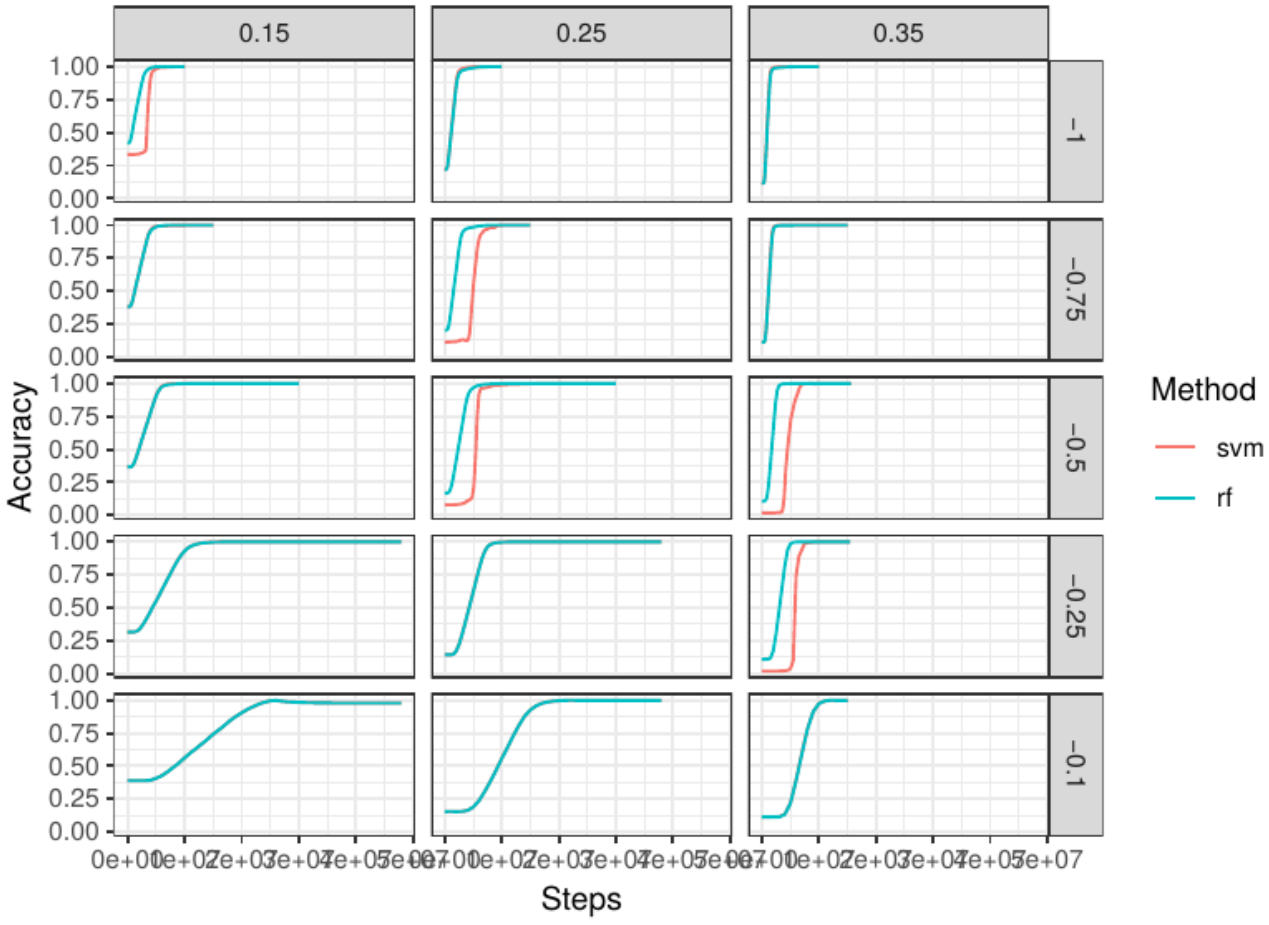
“script_rf_svm_small”
Utilizando lo de “prueba2”, automaticé la elección de step de corte y de clasificación de granos en el script que devuelve el accuracy para rf y svm que teníamos originalmente. En este script, cada simulación entrena con sus velocidades y ff respectivos y prueba en los archivos que corresponde a su propia v y ff.

En la elección del corte no hice la elección con el mismo criterio que en “script_corte_analistaR”. En este, la elección se hace simplemente con lag y que el corte suceda cuando la resta entre la energía de un step y el anterior sea 0.

plot antes de automatizar:



plot automatizado:

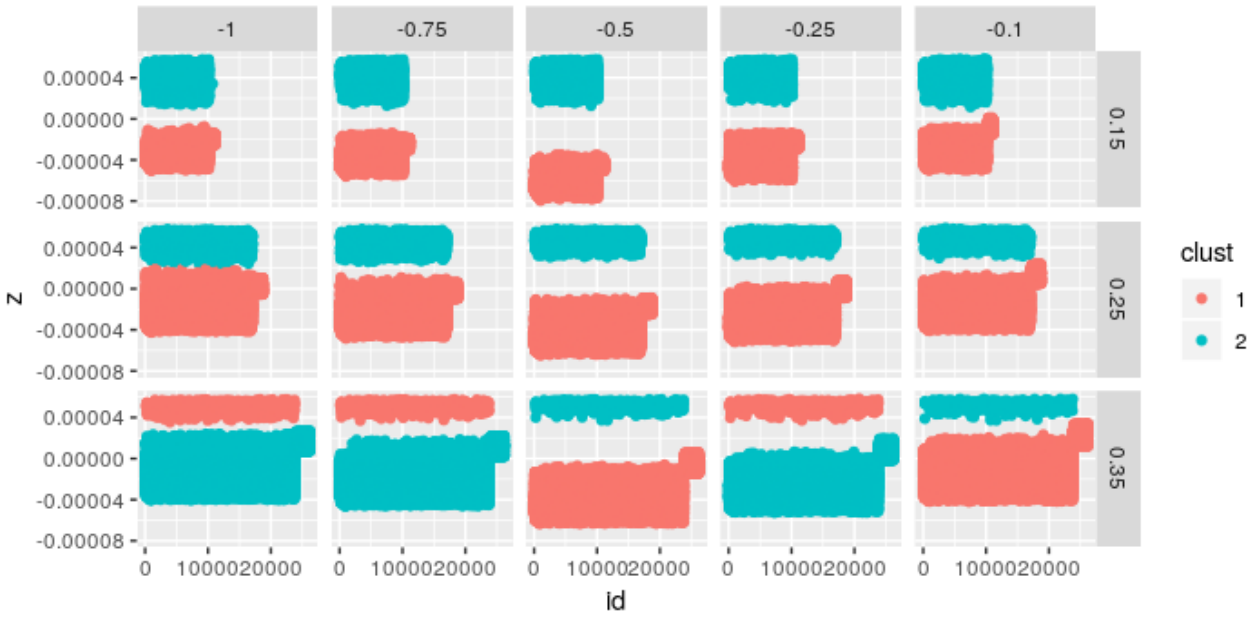


“super_training”
Hice un script que hace un df con todos los “últimos” outputs (según el criterio de corte) de cada simulación, clasificados por vel y ff, y con la división de pertenencia a cada cluster.
Filtré primero por las vel con ff 0.15 y entrené con ese df.
Utilicé ese df para evaluar de forma local, en velocidades con ff 0.15 (elegí vel 1 y después 0.1).
Dio bastante bien.

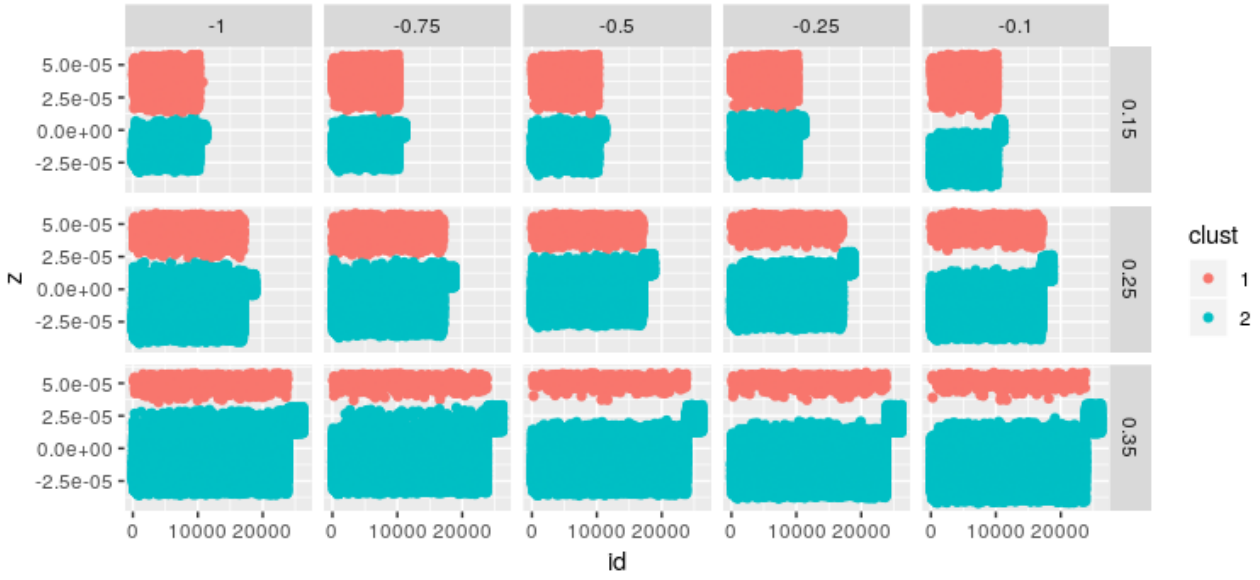
```
0.15, -0.1, 100000000

step:20000000
Accuracy : 0.3178
Sensitivity : 1.00000
Specificity : 0.02687

step 30000000
Accuracy : 0.984
Sensitivity : 0.9468
Specificity : 0.9999

step:50000000
Accuracy : 0.9783
Sensitivity : 0.9274
Specificity : 1.0000

step: 10000000
Accuracy : 0.9783
Sensitivity : 0.9274
Specificity : 1.0000


0.15, -0.5, 30000000

step: 5000000
Accuracy : 0.9058
Sensitivity : 1.0000
Specificity : 0.8697

step: 10000000
Accuracy : 0.9996
Sensitivity : 1.0000
Specificity : 0.9995

step:20000000
Accuracy : 1
Sensitivity : 1.0000
Specificity : 1.0000

step:30000000
Accuracy : 1
Sensitivity : 1.0000
Specificity : 1.0000


0.15, -1, 10000000

step: 2000000
Accuracy : 0.7779
Sensitivity : 0.9850
Specificity : 0.6742

step: 3000000
Accuracy : 0.9596
Sensitivity : 0.9858
Specificity : 0.9464

step: 5000000
```

```
Accuracy : 0.9916
Sensitivity : 0.9757
Specificity : 0.9996

step:10000000
Accuracy : 0.9984
Sensitivity : 0.9952
Specificity : 1.0000
```

Luego de realizar varias pruebas utilizando la energía como criterio de corte, nos dimos cuenta que no basta por sí solo para dar un corte confiable. Comparando los cortes sugeridos con los pasos en OVITO, nos damos cuenta que un analista nunca cortaría tan temprano la simulación, y para atrasar el corte se necesitan criterios más forzados (rayan en la prueba y error)
la idea es ahora:
-Tomar el último output de cada simulación
-Clasificar en clusters usando kmeans, respecto de la posición en z. Llegué hasta acá
-Encontrar el átomo del cluster 1 que esté más cercano al cluster 2 y calcular esa distancia.
-ir hacia atrás en los outputs: cuando esa distancia sea de 5 radios (elegido así por OVITO, para todos los outputs el mismo criterio), cortar la simulación.
- Problemas que veo antes de hacer esto: los puntos mas cercanos entre clusters pueden ir cambiando. ← SOLUCIONADO: esto no pasa porque para cada nuevo output en el que se mete la función me computa las partículas más cercanas en ese output.
- Plot mostrando clasificación por velocidad y posición, porque con posición solamente, clasificaba mezclado.



Nuevo archivo: super_lo
Este script toma cada directorio de las simulaciones, entra a cada uno, empieza desde el último output hacia atrás, a cada uno le aplica la clasificación de kmeans por z y vz hasta que la distancia entre la partícula del c1 más cercana a una partícula de c2 es de 5 radios. Cuando eso sucede se detiene. Me hace un super df de training con estos scripts de corte. También me devuelve un df que tiene en dónde este método corta la simulación y cuantos segundos habría tardado la simulación si se cortaba en ese step.



CON ESTE DF VOY A EXPLORAR EL ESPACIO DE PARÁMETROS.

	size	fill	velocity	last_step	final_t
1	small	0.15	-0.10	89120000	217814.63
2	small	0.15	-0.25	32330000	81451.56
3	small	0.15	-0.50	15520000	33982.33
4	small	0.15	-0.75	9900000	21649.19
5	small	0.15	-1.00	7330000	16738.71
6	small	0.25	-0.10	96320000	303522.62
7	small	0.25	-0.25	33050000	106162.55
8	small	0.25	-0.50	15250000	41746.67
9	small	0.25	-0.75	12050000	33926.21
10	small	0.25	-1.00	9800000	28916.86
11	small	0.35	-0.10	9800000	22236.49
12	small	0.35	-0.25	37740000	138803.95
13	small	0.35	-0.50	18070000	42105.57
14	small	0.35	-0.75	11800000	38487.27
15	small	0.35	-1.00	9000000	29084.31

Anotación:
estaba mal hecho el código, la tabla anterior muestra que hay valores de step de corte que se repiten y las pruebas daban cualquier cosa.
Lo solucioné y empecé a hacer pruebas. Guardé todos los resultados y agregué otras medidas de desempeño: kappa, recall y precision. Voy a explorar ROC y AUC para comparar rf y svm.
LAS PRUEBAS QUE VOY REALIZANDO ESTÁN EN EL NUEVO DOC “pruebas_”