**Formatos para diccionario metagenomas**

**Formato 1: unión de todos los reads en un pseudo-genoma**

La idea es encadenar todos los reads en un pseudo-genoma y tratar el caso como si fuera un GenomaVsGenoma.

* Pro(s):
  + No hay que modificar el código.
* Contra(s):
  + Se generan palabras virtualmente creadas.
  + Se obvian los fragmentos intermedios entre reads.
  + El fichero de metagenomas debe estar tratado y los reads en su posición correcta para que el resultado sea *fiable*.

El formato sería el básico:

**word \t pos**

**Formato 1.1: unión de todos los reads en un pseudo-genoma (mejorado)**

La idea es la misma que en el formato 1 pero añadiendo un campo que tenga en cuenta si la palabra en cuestión es una palabra virtualmente creada o no (palabra en unión de reads) para ello sólo habría que añadir un campo lógico al diccionario.

* Pro(s):
  + Casi no hay que modificar el código.
  + Permite elegir entre tener, o no, en cuenta las palabras virtualmente creadas lo que lo hace un pseudo *formato 2*.
* Contra(s):
  + El cálculo del diccionario necesita de tener en cuenta las posiciones de enlace de reads.

El formato sería:

**word \t pos \t BoolValue**

**Formato 2: tratamiento de reads como pseudo-genomas individuales**

La idea es tratar cada read del metagenoma como un genoma individual, esta idea se basa en el hecho de que desconocemos los fragmentos que hay entre read y read y por tanto sería acertado entender cada read como un *genoma* individual y buscar la inclusión que tiene con un genoma.

* Pro(s):
  + Facilidad para paralelizar.
  + No se obvian fragmentos del metagenoma desconocidos.
  + No hay que modificar casi nada el código.
* Contra(s):
  + Consume más recursos.
  + Hay que modificar el código para que identifique los diferentes reads.

El formato del diccionario sería:

**word \t readIndex \t pos**

Con este formato, para paralelizar sólo habría que crear diccionarios temporales para cada conjunto con igual *readIndex* y lanzarlos en paralelo.