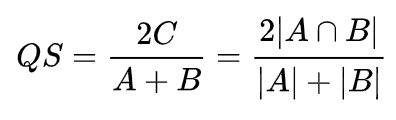
Aprendizaje semana 14/06/2021:

Coeficiente de Dice:

Muy similar al coeficiente de Jaccard, con algunas propiedades diferentes. La ceunta es un poo distinta:



Como se puede ver, en el denominador cuenta a los conjuntos de forma independiente, de modo de que si comparten elementos, estos van a ser contemplados 2 veces en el denomindor.

<https://es.wikipedia.org/wiki/Coeficiente_de_Sorensen-Dice>

Garnett:

Algoritmo de clasificación de células:

Cómo funciona:

Parte de un archivo .txt que el programa llama *marker\_file.* Este contiene los tipos de células esperadas y los marcadores conocidos de bibliografía correspondientes a cada uno de estos tipos. El archivo tiene esta pinta:

>B cells

expressed: CD19, MS4A1, CD79A

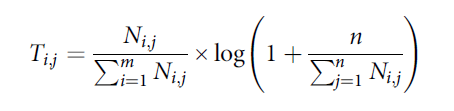
>T cells

expressed: CD3D, CD3E, CD3G

En donde expressed significa que con 1 count de esos marcadores, la célula podría ser clasificada dentro de alguna de esas categorías. Este clasificación es solo un aproximado y sirve para los pasos posteriores del algoritmo.

Con la función check\_markers(), usando como input el *marker\_file* y el perfil de expresión no normalizado, el programa clasifica a todas las células dentro de las posibles categorías. Con las células clasificadas, el programa puede determinar el número de células que fueron asignadas a cada categoría (*total*\_*nominated*) y las células que podrían potencialmente ser clasificadas a cada categoría si solo tuviésemos en cuenta uno de los marcadores de dicha categoría (*nominates*) (y esto es básicamente el número de células que expresan (count>=1) ese marcador). Como obviamente pueden haber marcadores de un tipo celular que estén expresados en células de otros tipos celulares, se puede establecer la ambigüedad (*ambiguity*) de los marcadores, es decir, que tambien esos marcadores ayudan a distinguir entre un subtipo celular respecto a otro. Esta ambigüedad está relacionada con el número de células que expresan dicho marcador + otros de una categoría celular + otros de otra categoría celular. Por último, por cada marcador se puede definir un *marker\_score* en base al nivel de expresión de éste en las células clasificadas dentro de la categoría del marcador.

El *marker\_score* por célulase puede definir como la expresión de ese gen respecto al resto de los genes en una determinada célula, modificado por la inversa de la expresión de ese gen en todas las células (si ese gen se expresa en solo algunas células, entonces es más específico y este término va a ser mayor), multiplicado por el número de células totales (cuanto más grande el dataset, más confianza en ese marcador).



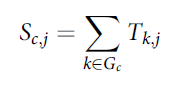
Acá Ti,j es la matriz de *marker\_scores*  por gen por célula, Ni;j es la matriz de expresión original, normalizada,  *m* es el número de genes totales y *n* es el número de células totales.

Para que un gen se considere expresado en un célula, se determina un cutoff de “expresión”, que está dado por la siguiente información:

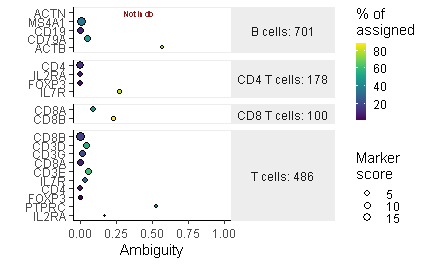


Donde Ci es el valor de cutoff y está dado por que 0.25% del valor de qi, que es el valor de expresión de un determinado gen i, que corresponde al percentil 95 de ese gen en la matriz Ti,j. Cualquier valor de expresión del gen i en la matrix Ti,j por debajo de Ci, es llevado a 0 (cero).

Finalmente se define el *marker\_score* agregado por célula (osea, independiente de un marcador específico) como la suma de los valores de la matriz T correspondiente a cada marcador del *marker-file (Gc)* para cada célula, del subtipo celular *c*:



Con toda esta información, se puede contruir un gráfico descriptivo, como el siguiente, que permita revisar el *marker\_file*  y determinar con criterio lógico y basado en los datos, cuales son los mejores marcadores de cada tipo y si es conveniente sacar alguno de ellos dentro de un tipo celular:



En este ejemplo, sería conveniente quitar el marcador PTPRC como marcador de las células T, ya que tiene un nivel de ambigüedad muy alto, y además porque se expresa en un % bajo de células T, se expresa poco (su score es bajo) y porque ya hay varios marcadores de células T per se. Con criterios similares también se podrían pensar en descartar a los marcadores ACTB e IL7R de las células B y CD4T, respectivamente. El CD8B no lo sacaría porque si no me quedaría con un único marcador para células CD8 y su ambigueda no es tan alta. Acá el Marker Score no se si es la sumar del *marker\_score* de cada gen para todas las células u qué, no queda muy claro.

Una vez corrida esta función y modificado el *marker\_file* la idea es ahora clasificar realmente a las células, sin embargo, en Garnett, las células no se clasifican solo con los marcadores elegidos, sino a partir de todos o una buena parte de los genes. Los marcadores del *marker\_file* modificdo nos sirven para encontrar las células más representativas de cada categoría y poder, con estas y su perfil transcripcional, generar un modelo que permita predecir la clasificación de todas las células de un dataset.

La generación del modelo predictor se encuentra dentro de la función train\_cell\_classifier(). Esta función hace 2 cosas: 1) A partir del *marker\_file* encuentra a las células más representativas de cada tipo y 2) “entrena” un modelo multinomial predictor que permitirá saber la clasificación de cada célula en base al perfil de expresión de cada una de estas. ¿Qué quiere decir que este modelo se entrena? Quiere decir que a partir de las células “más respresentativas” de cada clase, identificadas por el *marker\_file,* y la expresión de todos los genes en cada una de ellas, se identifican los coeficientes (los betas) de un modelo lineal generalizado, multinomial, para luego con esos betas poder predecir la clasificación de una célula cualquiera (no la más representativa) a partir de la expresión de sus genes.

Solo como un detalle, el modelo es un elastic-net regularized glm, es decir que es un glm penalizado con un parámetro lineal, como en la regresión lasso, pero regularizado a …¿?

Yo pienso al modelo multinomial como un modelo que incluye una función por categoría:

El algoritmo de Garnett está obligado a incluir siempre dentro de las posibles clasificaciones del modelo, un outgroup. Este outgroup va a estra conformado por cantidades iguales de células de cada tipo celular posible. La idea de tener un outgroup es que el algoritmo no incluya una célula que no pertence a ningún tipo previamente determinado solamente por “descarte” (recordar que los modelos multinomiales se manejan con probabilidades, donde siempre se elige la categoría con la probabilidad más alta, por más de que todas sean bajas). Esto sirve para poder encontrar nuevos tipos celular, no esperados inicialmente. Martín dice que si pudiese existir más de outgroup, algunos conteniendo un mayor % de un tipo celular que otros, podríamos encontrar tipos celulares de transición entre tipos celulares conocidos. A Garnett hay que fijarle automáticamente el número de células totales utilizado para definir el modelo correspondiente al outgroup. Para esto hay que tener en cuenta en número de células del dataset. Yo pienso que tendría sentido usar el 10% de las células del dataset para el outgroup. Pero creo que uno puede jugar entre el 5 y 20% de las células y ver como la predicción cambia.

Finalmente, una cosa interesante de Garnett es que se pueden definir inicialmente jerarquías de tipos celulares, osea, dentro de las células T, puedo tener la CD4 y las CD8, por ejemplo. En este tipo de casos el entrenamiento sevuelve a ejecutar dentro de ada nivel de la jerarquía. Por ejemplo: primero solo se entrena teniendo en cuenta la clasificación en células T y B. Luego se clasifican a las células T y dentro de este subset de células, se entrena otro modelo teniendo en cuenta la clasificación en CD8 y CD4 y así sucesivamente hasta llegar al nivel más bajo.

Otra de las curiosidades que tiene Garnett es que permite entrenar un modelo, con un dataset, para luego utilizarlo en un dataset nuevo, sin necesidad de repetir todo el análisis. Además, tiene funciones que permiten encontrar genes homólogos entre especies, por lo que permite clasificar a células de datasets que correspondan a especies distintas.

<https://www.nature.com/articles/s41592-019-0535-3>

Diferencias entre los distintos métodos de integración:

Todos los métodos de integración suponen que la estructura poblacional de los tipos celulares presentes en una muestra se mantiene visible dentro de cada batch. Es decir, que la variabilidad provocada por el batch no es mayor a la variabilidad provocada por el tipo celular.

Integración de Monocle 3:

No se modifica la matriz de datos original, solo se modifica el espacio dimensional reducido, para poder hacer el clustering de forma apropiada. Luego, en el cálculo de la expresión diferencial se incluye el término del batch. Entonces, en este caso, la corrección numérica de batch es gen dependiente, pero a su vez teniendo en cuenta a todas las células del batch por igual. Es decir, hay un único término de batch que aplica para todas las células de ese batch, para ese gen. Si el análisis de expresión diferencial lo hago con término de batch con interacción con el cluster, entonces el término de batch aplica a todas las células del batch de un determinado cluster por igual. Esto podría ser útil para comparar clusters de distintos tipos celulares, una situación muy común. Pros: contempla el caso en donde hay genes más sensibles al batch effect que otros. Cons: No contempla los casos en los que el efecto de batch hace que las células de un tipo estén más dispersas entre sí en un batch que en el otro (podría ser el caso en donde intento integrar experimentos diferentes o condiciones diferentes, en donde uno de ellos –por un efecto NO biológico- tiene clusters menos definidos que el otro).

Integración por Seurat:

Se modifica la matriz de datos original y se genera una nueva matriz con los datos corregidos. En la corrección de los datos se ajusta todo el perfil de una célula de un batch a su anchor correspondiente en el otro batch. Osea, es una corrección gen independiente, célula dependiente. Es decir que cada célula tiene una corrección diferente. Pros: contempla que cada tipo celular puede tener una sensibilidad distinta al batch. Cons: No contempla que cada gen puede tener una sensibilidad distinta al batch effect. Podría pasar, por ejemplo, que si el batch effect está dado por el paso de amplificación de la libraries mediante PCR, haya algunos genes se amplifiquen más rápido que otro debido a su longitud y que si de por sí hay diferencias de amplificación entre los batches, que hayan genes que se vean más afectados por estas diferencias que otros, por ejemplo, los genes más cortos. Esta corrección tiene sentido cuando se intentan integrar datos de técnicas totalmente distintas o de distintas condiciones experimentales. En estos casos capaz la variabilidad del efecto de batch entre genes no es tan grande como la variabilidad del efecto de batch entre células del mismo tipo.

Igualmente ninguno de estos métodos hace magia y si los datos entre batchs son muy distintos, tampoco va a poder superponerlos perfectamente.

El único método que hace un poco de over-correction es el de CCAs de Seurat, porque trabaja con “vectores de correlación” (después de mucha búsqueda aún no termino de entender bien que son) y no con los valores absolutos para hacer la reducción dimensional. Es por eso que el propio paquete recomienda usar RPCA en caso de tener datos que espero que sean muy disímiles, ya que RPCA construye los PCs a partir de los valores absolutos.

Ya después la aparición o desaparición de tipos celulares es un fenómeno biológico, batch independiente (o eso se asume si trabajo con condiciones experimentales distintas, réplicas biológicas, etc) y no debería ser corregido, de lo contrario, debería ser tenido en cuenta en ls análisis de expresión diferencial (y tanto Monocle3 como Seurat los tienen en cuenta).

Texto de CCA: <https://sci-hub.st/https://doi.org/10.1007/978-1-4939-7131-2_110191>

Paper comparativo de algoritmos de integración del single cell: <https://genomebiology.biomedcentral.com/articles/10.1186/s13059-019-1850-9#Sec14>

Este paper termina concluyendo qué (conclusiones + mi propia interpretación) los mejores métodos de integración son:

* Si tengo != tipos celulares en distintos batches:
  + scMerge:
    - Pero falla si los tipos celulares de un batch están incluidos en el otro.

tipos celulares del batch1: CMP, GMP, MEP

tipos celulares del batch2: CMP, GMP, MEP + otras

* + - Sirve para hacer DEG (Diferencial Expression of Genes) con los HVG (High Variable Genes) y da buenos resultados (F-score con TruePositives, False Positives, True Negatives y False Negatives)
    - Consume muchísima memoria y requiere mucho tiempo de ejecución el cual es independiente del número de células.
* Si tengo = tipo celular pero muchos batches (5 o +):
  + Seurat3 (solo analizaron el CCA):
    - Consume mucha memoria
    - Sirve para hacer DEG con los HVG y da buenos resultados (F-score)
  + Harmony (corrido desde Seurat):
    - Consume poca memoria
    - No sire para hacer DEG, no devuelve una matriz corregida.
* Si tengo = tipo celular pero datasets muy grandes:
  + Seurat3 (solo analizaron el CCA):
    - Consume mucha memoria
    - Sirve para hacer DEG con los HVG y da buenos resultados (F-score).

<https://genomebiology.biomedcentral.com/articles/10.1186/s13059-019-1850-9#Sec14>

F-score: <https://en.wikipedia.org/wiki/F-score>