
FACULTAD DE CIENCIAS

DEPARTAMENTO DE MATEMÁTICAS
PROYECTO FINAL DIPLOMATURA DE ESTADÍSTICA

# Modelización de conteos mediante la distribución Poisson-Tweedie (PT): aplicación en datos de ultrasecuenciación.

Proyecto presentado por Manuel Carrasco Peña (1215173)

Tutorizado por:
Juan Ramón González, Investigador en el Centro de Investigación en Epidemiología Ambiental (CREAL)

# Índice general

# Índice de figuras

# Sección 1

# Introducción

Generalmente, en investigación se usa la estadística y en especial los modelos probabilísticos como la herramienta primordial para hacer inferencia sobre una base de datos, e intentar extraer conclusiones. Un método adequado de análisis juega un papel fundamental en la toma de decisiones en pro de la solución de los problemas que se abordan.

Por otra parte, el análisis de datos de conteo es de uso muy habitual, asi podemos encontrar ejemplos de este tipo de datos, en epidemiologia,(número de hospitalizaciones, número de dolores de cabeza, número de casos detectados de una enfermedad, etc), biologia (número de celulas rojas en un tejido, número de especies en un entorno), genética (número de veces que se expresa un gen), en la industria (número de averias de una máquina) y prácticamente en cualquier ambito encontramos datos de este tipo. Nosotros prestaremos especial atención a datos de tipo genómico donde se cuentan la veces que se expresa un gen en el genoma.

En los modelos de conteo, la variable dependiente es discreta y no negativa, además se caracterizan porque no tienen un límite superior natural, toman el valor cero (en un porcentaje no despreciable), para algunos miembros de la población y suelen tomar pocos valores. Para modelizar este tipo de datos la opción clásica es la de utilizar el modelo *Poisson*, tratando la componente aleatoria de los datos según las características de esta distribución. Sin embargo, son conocidas las limitaciones de este enfoque que menosprecian el supuesto restrictivo de que la media y la varianza para la distribución Poisson deben ser iguales. En datos reales esta situación raramente se encuentra y normalmente los datos presentan sobredispersión, es decir la variancia de los datos es más grande que la media, entre otros problemas. Por esta razón se han planteado diferentes modelos que intentan recoger la posible sobredispersión de los datos.

Los modelos basados en la distribución Binomial Negativa incorporan una componente de variabilidad que hace que datos con sobredispersión sean modelizados con mejores resultados que con el modelo Poisson, ya que la Binomial Negativa tiene un parámetro que controla la sobredispersión.

Otro problema habitual en datos de conteo es el exceso de ceros. En este caso se usan modelos Cero Inflados y Hurdle, basados en la distribución Poisson o Binomial Negativa. Estos modelos son una mixtura que consideran que para una proporción de individuos se observan ceros, y para el resto los datos provienen de una Poisson o una Binomial negativa. Este tipo de modelos presentan limitaciones evidentes, como es el hecho que tengan que presentrar como mínimo algún cero. El hecho de usar mixturas de distribuciones en las que se realiza inferencia de forma independiente (por ejemplo, ver si la proporción de ceros es igual en ambos grupos) puede ser una limitación en algunos contextos, como es el caso de datos genéticos, como veremos mas adelante, donde observar ceros nos puede indicar que un gen no se expresa, pero el investigador no esta interesado en modelar aquellos

individuos con ceros y sin ceros de forma independiente.

En definitiva, existen distintos modelos probabilísticos para analizar datos de conteos. Para cada problema debe buscarse el mejor de ellos para llevar a cabo los análisis que ayuden a testar la hipótesis del investigador. Sin embargo, como veremos a continuación, en genómica hay casos en los que se deben analizar miles de variables de este tipo por lo que la elección de un único modelo para todos los datos se hace imposible. Es por ello que en este proyecto presentaremos la distribución, Poisson-Tweedie (PT), que dada su flexibilidad para modelar datos de conteos será una distribución muy adecuada para modelar datos de RNA-seq obtenidos mediante ultrasecuenciación, ya que cada uno de los genes analizados puede requerir de un tipo distinto de distribución para ser analizado. A continuación detallamos el porqué del uso de esta distribución, asi como el problema que ha motivado el trabajo.

# Sección 2

# Motivación: Modelado de datos de RNA-seq para datos de ultrasecuenciación

La secuenciación de ADN es un conjunto de métodos y técnicas bioquímicas cuya finalidad es la determinación del orden de los nucleótidos (A, C, G y T) en un oligonucleótido de ADN. La secuencia de ADN constituye la información genética heredable del núcleo celular, los plásmidos, la mitocondria y cloroplastos (en plantas) que forman la base de los programas de desarrollo de los seres vivos. Así pues, determinar la secuencia de ADN es útil en el estudio de la investigación básica de los procesos biológicos fundamentales, así como en campos aplicados, como la investigación forense [3].

El desarrollo de la secuenciación del ADN ha acelerado significativamente la investigación y los descubrimientos en biología. Las técnicas actuales permiten realizar esta secuenciación a gran velocidad, lo cual ha sido de gran importancia para proyectos de secuenciación a gran escala como el Proyecto Genoma Humano. Otros proyectos relacionados, en ocasiones fruto de la colaboración de científicos a escala mundial, han establecido la secuencia completa de ADN de muchos genomas de animales, plantas y microorganismos. El uso de el análisis genético tiene múltiples aplicaciones en medicina, y es un campo que tiene mucha proyección en diversas áreas médicas, tales como la detección prematura de enfermedades o la aplicación de tratamientos individualizados, a partir del análisis genético del individuo.

La ultrasecuenciación es una de las nuevas técnicas de secuenciación masiva de ADN (también conocidas como Next generation Sequencing). Estas son técnicas que permiten la secuenciación de mayor número de nucleótidos en comparación con la métodos basados en la técnica de Sanger, que es considerada la técnica de referencia para secuenciación de ADN [6].

El RNA Sequencing (RNA-Seq), es una tecnologia que utiliza la ultrasecuenciación para detectar y cuantificar la cantidad de ADN de un génoma en un determinado momento del tiempo. En la figura 2.1 podemos ver un esquema del funcionamiento de esta tecnología y qué tipo de datos se obtienen. En primer lugar, y dado que es muy complicado secuenciar el genoma completo debido a su gran tamaño, se fragmenta el ADN en pequeños trozos que se conocen como ”reads”, en inglés, estos ”reads” se secuencian y finalmente se alinean contra un genoma de referencia estandard. Ese alineamento se puede llevar a cabo frente a distintas partes del genoma (exones, transcritos, genes, ...), *Chu Y et al.* (*August* 2012).

En el caso del RNA-seq como interesa contabilizar el nivel de expresión génica, se alinean contra genes, de forma que si para un individuo un gen expresa más que otro, eso implicará que el número de 'reads' alineados contra ese gen seran mayor que contra el otro. Al final de este proceso, y para cada individuo, se obtiene una matriz de conteos para cada uno de los genes (ver Tabla 2.1). Normalmente se obtiene información para unos 20.000 genes. En este tipo de estudios, también se recoge información para dos grupos de individuos o condiciones, en los que se pretende determinar qué genes están diferencialmente expresados entre grupos, es decir, qué genes tienen una media en el número de conteos distinta para cada condición.

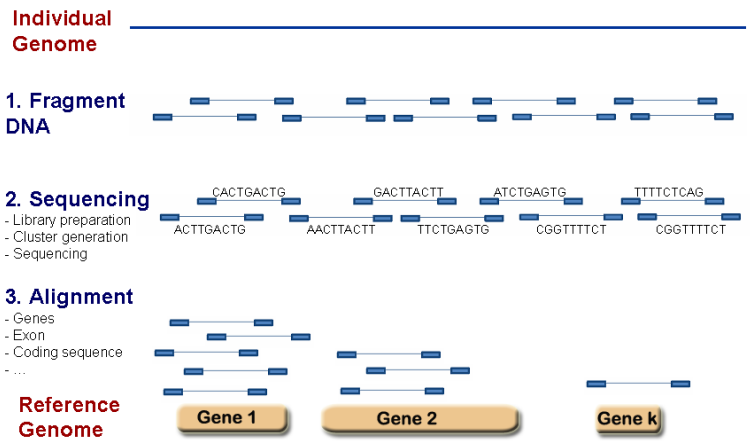


Figura 2.1: Esquema del sistema de ultrasecuenciación para ADN, RNA-seq, observamos en primer lugar la fragmentación el ADN, en segundo lugar se secuencian los fragmentos y en tercer lugar estos fragmentos son alineados con un gen de referencia para producir una tabla de conteos

| | Grupo A | | | | Grupo B | | | |
|---|---|---|---|---|---|---|---|---|
| | Individuo 1 | Individuo 2 | ... | Individuo k | Individuo k+1 | Individuo k+2 | ... | Individuo n |
| Gen 1 | 400 | 12 | ... | 2 | 3 | 37 | ... | 0 |
| Gen 2 | 16 | 22 | ... | 8 | 27 | 16 | ... | 4 |
| Gen 3 | 473 | 437 | ... | 327 | 607 | 199 | ... | 128 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| Gen n | 2 | 2 | ... | 0 | 0 | 0 | ... | 3 |

Tabla 2.1: Matriz de datos obtenida mediante RNA-seq. El objetivo es comparar la expresión génica para cada gen entre el grupo A y B. Para ello se requiere del uso de una distribución lo suficientemente flexible para que sea capaz de modelar tanto la sobredispesión, como el exceso de ceros o colas pesadas

Si nuestro interés es analizar un único gen, este problema está bien resuelto ya que los datos se pueden modelar con cualquiera de las distribuciones que anteriormente hemos citado. Sin embargo, si estamos interesados en analizar miles de genes, deberíamos estudiar qué modelo es el más adecuado para cada uno de ellos. Además, en el caso de este tipo de estudios el uso de modelos Cero Inflados no es adecuado desde un punto de vista biológico ya que en genética los investigadores no les interesa saber si el porcentaje de ceros que hay entre dos condiciones es distinta o no. Les interesa saber si en promedio la expresión de un gen está alterada. Además, los investigadores también han notado que a veces hay individuos con conteos mucho más grandes de lo normal, que hace que aparezca otro problema añadido, cuando se analizan datos de conteos, como es el de las colas

pesadas que los modelos tradicionales no contemplan [10].

Por todo ello, el uso de modelos Poisson o Binomial negativo, nos pueden dar resultados erróneos debido al mal ajuste de los datos pudiendo dar como resultado falsos positivos. El uso de modelos del tipo Hurdle y cero inflados también es limitado y esta condicionado a la existencia de un exceso de ceros, cosa que en muchos casos no se observa. Por tanto el uso de modelos con una distribución más generalizada, como los modelos Poisson-Tweedie (PT) aportan una ventaja clara ya que incluyen un mayor número de posibles distribuciones, y en general nos aportan un mejor ajuste, ya que son modelos capaces de adaptarse tanto a datos con exceso de ceros, presencia de sobredispersión o con colas pesadas. En este trabajo mostraremos que la distribución PT es más apropiada para analizar datos de ultrasecuenciación, como son los obtenidos por la técnica de RNA-seq.

## 2.1. Datos reales

Los datos que han motivado el presente trabajo proviene de los datos obtenidos mediante RNA-seq que fueron generados por *Pickrell et al.* (2010) que secuenciaron RNA de células linfoblastoides en 69 Nigerianos. El proceso de generación de datos fue el mismo que el ilustrado en la Figura 2.1 y la tabla final de conteo de datos está accesible en el paquete `tweeDEseqCountData` que está accesible en `http://www.bioconductor.org` con el nombre `pickrell1Norm`. El total de genes disponible es de 22.060. Para este trabajo, y para evitar que el coste computacional no sea una limitación, se ha seleccionado una serie de 15 genes para ilustrar los análisis y algunas de las simulaciones con datos reales.

# Sección 3

# Objetivos

El principal objetivo de este trabajo es estudiar el comportamiento del modelo PT en el análisis de datos de conteos que aparecen en los estudios genómicos. En particular, nos centraremos en el análisis de datos de RNA-seq obtenidos mediante ultrasecuenciación. Para ello se llevarán a cabo tanto estudios de simulación como análisis de datos reales en los que se comparará el comportamiento de esta familia de distribuciones con otras distribuciones normalmente usadas para analizar este tipo de datos como son, Poisson, Binomial Negativa (incluyendo sus versiones cero infladas y Hurdle). El objetivo final, es poder definir el mejor modelo para comparar dos grupos de individuos y así poder determinar qué genes están diferencialmente expresados.

Los estudios de simulación se basarán en:

- Analizar datos sintéticos generados a partir de distribuciones conocidas en las que se variarán los parámetros de las distribuciones simuladas para cubrir distintos escenarios. El comportamiento de cada modelo se valorará en función de si es capaz o no de controlar correctamente el error de tipo I. Los modelos que controles este error de forma correcta se compararán mediante estudios de potencia y criterios de bondad de ajuste para el modelado de datos (AIC).

- Para evitar posibles influencias de obtener mejores resultados de un modelo según se hayan simulado los datos, se utilizará un conjunto de datos reales en los que se evaluará el error de tipo I para cada modelo mediante simulación en el que se permutará el grupo de comparación de forma que se pueda simular datos bajo la hipótesis nula que no haya diferencias en la media de conteos entre ambos grupos.

Por otro lado, el análisis de datos reales consistirá en usar todos los modelos existentes para comparar la expresión génica, medida mediante datos de RNA-seq de 69 individuos en los que se pretende estudiar si hay diferencias en cuanto a la expresión génica entre hombres y mujeres. Para evaluar qué modelo es el más adecuado para analizar cada gen utilizaremos dos criterios:

- Un criterio estadístico basado en medidas de bondad de ajuste y pruebas de hipótesis.

- Puesto que en la literatura biomédica se conoce qué genes están diferencialmente expresados entre hombres y mujeres, los usaremos para determinar si el análisis mediante cada modelo determina de forma estadísticamente significativa si la media de conteos para genes es distinta entre ambos grupos de comparación. En definitiva, se dispondrá de un ”gold estandard” para evaluar cada uno de los métodos analizados.

# Sección 4

# Descripción de los modelos existentes para datos de conteo

- **Regresión Poisson**

  En este modelo se asume que la distribución de la variable de interés sigue una distribución de *Poisson*, $D \sim Pois(\lambda)$.

  La función de densidad de la variable respuesta es:

  $$Pr(Y = y|\lambda) = \frac{e^{-\lambda}\lambda^{y}}{y!}, \qquad para \;\; y = 0, 1, 2, ...$$

  Si la especificación para la distribución condicional de la variable respuesta, así como la de la media condicional, es correcta, y bajo el supuesto de que se tienen observaciones independientes, entonces se puede utilizar la siguiente función de verosimilitud para obtener estimadores consistentes de $\beta$:

  $$L(\beta|y, X) = \prod_{i=1}^{N} Pr(y_i|\mu_i) = \prod_{i=1}^{N} \frac{e^{-\mu_i}\mu_i^{y_i}}{y!}$$

  Donde

  $$\mu_i = E(y|x_i) = e^{(x_i\beta)}$$

  El método de estimación a seguir, es el de máxima verosimilitud, lo que podria conducir a estimadores inconsistentes si la función de probabilidad no está bien especificada [4].

  En la literatura hay datos de conteos correctamente modelados con la ley Poisson, pero en otros casos no se cumple el supuesto de equidispersión y el ajuste Poisson no es adecuado. Esta situación se debe a que en algunas ocasiones hay correlación entre las observaciones, falta de homogeneidad en el material experimental, existencia de datos atípicos o exceso de ceros, factores, todos estos, que generan un fenómeno llamado sobredispersión. La falla del supuesto de equidispersión en el modelo Poisson tiene consecuencias cualitativas similares a la falta de homocedasticidad en el modelo de regresión lineal, pero la magnitud del efecto sobre los errores estándares de los estimadores de los parámetros del modelo puede ser mucho mayor. La sobredispersión se presenta cuando la variación estimada es más grande que la predicha para el modelo, lo cual implica la sobrestimación de los errores estándar derivando en la subestimación de la significancia estadística de los coeficientes de los parámetros, llevando a que algunos coeficientes sean considerados significativos cuando realmente no los son, por lo tanto pueden derivar en la aplicación de

intervenciones sobre factores que verdaderamente no sean infuyentes [2].

- **Regresión Binomial negativa**

Una de las principales razones por las que el modelo Poisson falla es la heterogeneidad no observada. Esto significa que hay factores no observados, en especial características de los individuos, que ejercen alguna influencia sobre la variabilidad relacionada con la variable de respuesta.

El problema es que la heterogeneidad no observada puede tener algunas consecuencias para los procesos de inferencia estadística. Esta heterogeneidad, ignorada por el modelo Poisson, puede modelarse de manera explícita mediante el uso de la regresión Binomial Negativa. Segun el modelo binomial negativo la variable respuesta se distribuye segun la función de densidad:

$$f(y;\mu,\theta) = \frac{\Gamma(y+\theta)}{\Gamma(\theta)\cdot y!}\cdot\frac{\mu^y\theta^\theta}{(\mu+\theta)^{y+\theta}} \qquad [4]$$

La distribucion Binomial Negativa tiene dos parámetros, $\mu$ y $\theta$. El parámetro $\mu$ corresponde a la media de la distribución y $\theta$ es el parámetro de sobredispersión, en el caso concreto en que  es igual a 0, nos encontramos que es equivalente a una distribución Poisson.

La función de verosimilitud del modelo binomial negativo es:

$$L(\beta|y,X) = \prod_{i=1}^{N} Pr(y_i|x_i) = \prod_{i=1}^{N} \frac{\Gamma(y+\alpha^{-1})}{y!\Gamma(\alpha^{-1})}\left(\frac{\alpha-1}{\alpha-1+\mu_i}\right)^{\alpha^{-1}}\left(\frac{\mu_i}{\alpha^{-1}+\mu_i}\right)^{y_i}, \quad y=0,1,2,... \qquad [2]$$

Por tanto podriamos resumir, que la distribución Binomial Negativa nos permite analizar con un mayor ajuste datos de conteo que presenten sobredispersión, ya que hay un parámetro que intenta recoger esta información.

- **Regresión Cero Inflada**

Algunas variables de recuento muestran un porcentaje de ceros muy grande. Esa cantidad de ceros no es consistente con las distribuciones Poisson o Binomial Negativa (generalmente es mayor).

Una forma de tratar datos con exceso de ceros son modelos tipo Cero Inflado(ZI), son modelos en los cuales se supone que que los ceros de los datos se generan en dos procesos separados, unos que son ceros absolutos y otros que provienen de una distribución Poisson o Binomial Negativa. Por tanto la probabilidad de que $y=0$ tiene dos componentes:

$$Pr(y_i=0) = g_i + (1-g_i)f(0)$$

Donde f es la función de probabilidad de la Binomial Negativa o de la Poisson y la probabilidad de que $y>0$:

$$Pr(y_i=j) = (1-g_i)f(y) \quad j>0 \qquad [4]$$

- **Regresión Hurdle**

Los modelos de tipo Hurdle también se basan en la suposición que los datos provienen de dos fuentes, los ceros y los datos positivos, que son analizados de manera separada. La idea básica es que hay una decisión

binaria que determina si el resultado es cero o no cero y una segunda parte de la decisión que determina los valores mayores que cero, cuando esa valla ('Hurdle'), cero, no cero, se ha cruzado. Se supone que ninguno de los ceros provienen de la distribución con la que son tratados los datos positivos, de esta manera se construye un modelo truncado por la izquierda en $y = 1$ para los datos de conteo, y otro modelo truncado por la derecha en $y = 1$ que modela los ceros exclusivamente. [2]

Formalmente sería:

$$f_{hurdle}(y; x, z, \beta, \gamma) = \begin{cases} \text{si } y = 0 & f_{cero}(0; z, \gamma) \\ \text{si } y > 0 & (1 - f_{cero}(0; z, \gamma)) \cdot f_{count}(y; x, \beta)/1 - f_{count}(0; x, \beta)) \end{cases}$$

En el supuesto de que los datos realmente presenten un exceso de ceros los modelos Cero Inflados y Hurdle consiguen buenos resultados, y estimaciones ajustadas, pero la interpretación de estos modelos se complica ya que se producen dos modelos, un modelo para los ceros y otro modelo para los datos positivos, duplicidad que nos incrementa la interpretabilidad del modelo.

Podemos encontrar más información de la manera de estimar los modelos mediante R, en el documento "Regression Models for Count Data in R" que podemos encontrar en la dirección `http://cran.r-project.org/web/packages/pscl/vignettes/countreg.pdf`.

- **Regresión Poisson Tweedie(PT)**

El comportamiento de este tipo de modelos será el objetivo de nuestro trabajo y sera objeto de comparación con el resto de modelos anteriormente expuestos.

El modelo de regresión PT se basa en la familia de distribuciones PT como modelo estadístico para datos de conteo. Este tipo de distribuciones han sido estudiadas por diferentes autores y trata de unificar diferentes distribuciones de datos de conteo como son la Poisson y la Binomial Negativa, Poisson-Inversa Gaussiana, Polya-Aeppli o Neyman type A. *El − Shaarawi et al.*, 2011.

La familia de distribuciones PT han sido redescubierta varias veces. *Gerber* (1991) la llama distribución *Binomial Negativa generalizada* y la define como una mezcla de Poisson con una distribución gamma generalizada, *Gerber* menciona que también aparece en la tesis de $Hofmann$ (1955). $Hougaard$ *et al.* (1997) la llaman P-G y la define como una mixtura de una Poisson con una distribución de la familia power-variance (GAMMA). *Kokonendji et al.* (2004) la llaman Poisson-Tweedie, y en su sección 11.1.2,
*Johnson et al.* (2005) tienen la misma parametrización como $Hougaard$ *et al.* (1997) y la llaman la familia Tweedie-Poisson, porque la gamma generalizada, distribución definida por Gerber (1991) y la distribución de potencia-varianza en Hougaard et al. (1997) eran en realidad la sub-familia de Tweedie (1947). Para $0 < a < 1$, *Zhu y Joe* (2009), a través del uso de la geometría por ponderación y los operadores de suma de Poisson-stopped, deriva en PT(a, ab, c), pero la derivación no se extiende a=0.( *El − Shaarawi et al.*, 2011).

Este tipo de distribuciones se pueden definir a través de una función de probabilidad que puede ser calculada a través de un algoritmo recursivo. *El − Shaarawi et al*(2011) comparó diferentes métodos y parametrizaciones de este tipo de distribuciones y consiguió un algoritmo muy rápido para calcular la función de probabilidad de las PT. En R encontramos la libreria de R `tweeDEseq` que implementa la función `glmPT()`, esta será la función que usaremos para estimar los parámetros de nuestro modelo. La función `glmPT()`, implementa el algoritmo recursivo propuesto por *El − Shaarawi*.

Una variable aleatoria que sigue un distribución PT, $Y \sim PT\,(a, b, c)$, tiene un vector de parámetros

$\theta = (a, b, c)^T$, definidos en el dominio:

$$\Theta = (-\infty, 1] \times (0, +\infty) \times [0, 1),$$

Una variable aleatoria PT tiene una función de probabilidad de la forma:

$$G_Y(y \mid a, b, c) = exp\left\{\frac{b}{a}[(1-c)^a - (1-cy)^a]\right\}, \quad (a, b, c)^T \in \Theta$$

Cuando $a \neq 0$, si $a = 0$:

$$\lim_{a \to 0} G_Y(y \mid a, b, c) = \left[\tfrac{(1-c)}{(1-cy)}\right]^b$$

Usando esta parametrización, el siguiente algoritmo recursivo se puede usar para calcular la función de probabilidad de una PT:

$$p_0 = \begin{cases} e^{b[(1-c)^a - 1]/a} & , a \neq 0, \\ (1-c)^b & , a = 0 \end{cases}$$

$$p_1 = bcp_0, \quad p_{k+1} = \frac{1}{k+1}\left(bcp_k + \sum_{j=1}^{k} jr_{k+1-j}p_j\right), \quad k = 1, 2, ...$$

donde

$$r_1 = \big(1-a\big)c, \quad r_{j+1} = \big(\frac{j-1+a}{j+1}\big)cr_j, \quad j = 1, 2, ...$$

y $p_i$ denota la probabilidad de observar $''i''$ conteos.$El - Shaarawi\ et\ al.$, 2011.

Para poder ser interpretado, se parametriza $\Theta = \big(a, b, c\big)$ como $\big(\mu, \phi, a\big)$ donde $\mu$ es la media, $\phi = \sigma^2/\mu$ es el índice de dispersión, ($\sigma^2$ es la variancia) y $a$ es el parámetro de forma que se usa para para definir las diferentes distribuciones (Poisson, Binomial negativa,..), como casos particulares de la PT.

La relación entre las dos parametrizaciones es:

$$c = \frac{\phi - 1}{\phi - a}, \quad b = \frac{\mu\big(1-a\big)^{\big(1-a\big)}}{\big(\phi-1\big)\big(\phi-a\big)^{-a}}$$

La distribución PT incluye diferentes distribuciones dependiendo del valor del parámetro a. Con a=1, tenemos una distribución *Poisson*, con $a = 0$, la Binomial Negativa (NB), con $a = 0{,}5$ , la Poisson Inverse Gaussian (PIG), con $a = -1$, la Pólya-Aeppli (PA), y la distribución Neyman type A con $a = -\infty$.

De esta manera es capaz de unificar diferentes distribuciones que se aplican en datos de conteo que presentan sobredispesión, como son las distribuciones NB y PIG, y también suponemos que tendrá un buen comportamiento en datos con exceso de ceros ya que los tres parámetros de la distribución PT cubren una amplia gama de tipos de datos que tienen datos extremos que hace que la distribución presente una cola pesada, y de datos con sobredispersión. Esta unificación nos proporciona, una simplificación a la hora de elegir el tipo de modelo a utilizar, cuestión que siempre presenta una dificultad añadida cuando se realiza un estudio estadístico. Gracias a esta flexibilidad en los modelos PT, hace que sea capaz de modelar diferentes escenarios, entre ellos, los que se presentan en el análisis de sequencias de RNA, ya que este tipo

de datos tienen unos rangos que pueden llegar a ser muy amplios y presentar colas en las distribuciones muy pesadas.

Esta simplificación es muy importante cuando el objeto del estudio no son una sola serie de datos, sino que son centenares o miles de series que imposibilita el análisis individualizado de cada serie, lo que nos lleva a que el uso de un modelo más general sea la opción más eficiente. Podemos encontrar una descripción más detallada de las características de la familia de distribuciones PT en el documento de $El - Shaarawi\ et\ al.$ (2011)

# Sección 5

# Simulaciones: Método

Hemos visto que la distribución PT es una familia de distribuciones que es capaz de modelar datos de conteos con distintas peculiaridades (sobredispersión, colas pesadas, exceso de ceros, ...) sin tener que prefijar cual de ellas es la más adecuada en cada caso ya que el parámetro de forma $a$ se estima con los datos. Cabe esperar, entonces, que esta característica puede resultar esencial para analizar los datos de RNA-seq en el que se analizan miles de conteos de genes, que cuantifican como se expresa cada gen, cada uno con una distribución distinta.

Este trabajo, pues, pretende verificar mediante simulaciones y el análisis de datos reales, cómo se comportan los modelos PT y si realmente aportan una mejora respecto a los modelos usados comunmente para el análisis de conteos como el Poisson, Binomial Negativo, o sus versiones Hurdle y con Ceros Inflados.

## 5.1. Simulaciones con datos sintéticos

En primer lugar realizamos simulaciones en las que los datos de la variable respuesta son datos aleatorios de una distribución escogida, estos datos los generamos utilizando las funciones que hay en R para tal efecto .

Mediante una función creada a tal efecto, generamos las bases de datos sobre las que haremos los modelos. Esta función, que adjuntamos en el anexo, nos crea una base de datos con el sesgo deseado. Para obtener los estadísticos y guardar los datos también creamos una función, que adjuntamos en el anexo. Esta función, consiste, básicamente, en un bucle que va generando los modelos y vamos guardando los p-valores de significación y los AIC para después construir las gráficas y crear las tablas.

Las simulaciones para contrastar la potencia de los modelos, las realizamos, también, a través de una función de R creada a tal efecto(ver anexo). Esta función genera dos tipos de datos con medias diferentes y también un factor que nos indica a que serie pertenece cada dato, como resultado obtenemos el porcentaje de p-valores significativos para cada una de las diferencias entre medias. Realizamos los modelos, recogemos el porcentaje de p-valores significativos y generamos las curvas de potencia para cada uno de los modelos.

### 5.1.1. Error de Tipo I

En primer lugar, intentaremos mediante simulaciones ver como actuan los modelos PT, en datos que presentan un exceso de ceros. Con este objetivo creamos series de datos con un exceso de ceros para modelar a partir de

ellos. Las distribuciones que usamos para las simulaciones son, pues, una mixtura de ceros y otra distribución (Poisson o Binomial Negativa).

$$f(x) = wI_{\{x=0\}} + (1-w)f(x)$$

Creamos una función en R, la cual nos permite crear los datos con el exceso de ceros deseado y también generamos un factor compuesto por unos y ceros correlativos (0,1,0,...), este factor nos servirá para comparar la capacidad de los diferentes modelos para detectar que el factor no es significativo y que, los datos y el factor, son independientes, serán pues unas simulaciones con las cuales esperamos comparar las especificidad de los modelos. La especificidad de un modelo estadístico es la probabilidad de que el modelo tenga resultados significativos y en realidad no lo sea, es decir la probabilidad de que el modelo de falsos positivos.

A partir de estos datos hacemos los modelos en función del factor, de esta manera los p-valores de significación del factor deberían de estar distribuidos uniformemente en el intervalo (0,1), si nuestro modelo funciona correctamente. Bajo estas condiciones, al hacer las gráficas de la función de distribución empírica de los p-valores (gráficas ECDF), deberíamos de observar una diagonal, si el modelo estima bien los datos, y no deberíamos de encontrar mas de un 5 % de los p-valores significativos ($< 0{,}05$).

Comparando las distribuciones empíricas de los p-valores obtenidos de los diversos modelos, podremos comparar el modelo construido en base a la PT, con los modelos, Hurdle y Cero Inflado , y ver si en datos con un exceso de ceros se estiman también correctamente con modelos basados en la distribución PT.

Otro estadístico que obtenemos para comparar los modelos será el AIC, por tanto, también obtenemos los AICs de todos los modelos para compararlos. Los AIC nos daran una idea de la calidad relativa de los modelos. El AIC es un estadístico que nos valora el equilibrio entre la complejidad del modelo y la bondad de ajuste de nuestros modelos. Será pués un estadístico a tener en cuenta para comparar modelos.

Para hacer los modelos hemos escojido, en el caso de las muestras con un exceso de ceros, una serie de datos de longitud 1000. De estos 1000 datos introducimos el exceso de ceros en la propoción adequada en cada caso de forma aleatoria utilizando la función `sample`, y creamos una base de datos de 500 series. Estas serán nuestras variables respuesta. Con estas 500 series realizamos los modelos con el factor , creado independientemente, como covariable explicativa, guardamos los resultados de los p-valores y AICs para poder estudiarlos y graficarlos, ya que será la mejor forma de visualizar los resultados.

Las bases de datos simuladas que hemos utilizado són de dos tipos, las creadas a partir de una Binomial Negativa, y las creadas a partir de una Poisson, en ambos casos hemos creado datos con diferentes parámetros y diferentes proporciones de ceros. En concreto hemos hecho las simulaciones con la distribución Poisson con media 2 y media 20, para cada media hemos simulado datos con 10 %, 20 %, 30 %, 40 %, y el 50 %, como porcentajes de ceros introducidos de más.

Los datos creados a partir de la Binomial Negativa hemos generado datos con media 2 y 20, y 1.5 y 2 para el parámetro de dispersión. Cada combinación de estos valores hemos probado con 10 %, 20 %, 30 %, 40 %, y el 50 % como porcentajes de exceso de ceros, al igual que en el caso de las simulaciones con datos Poisson.

De los resultados obtenidos, destacamos que en muchos de los casos estudiados, observamos que con un exceso de ceros las los modelos del tipo Hurdle y Cero inflado actuan adequadamente, como era de esperar. La PT también se comporta de la manera correcta obteniendo en muchos de los casos, resultados que se ajustan a lo que esperabamos, es decir, los p-valores de los tres modelos se distribuyen uniformemente. Este hecho lo podemos observar adequadamente en las gráficas de distribución empírica que realizamos (ECDF).

Por otra parte las distribuciones Binomial Negativa, y Poisson vemos que en algunos casos no estiman correctamente los datos. Se observa que en la mayoria de casos, los p-valores obtenidos no se ajustan a una distribución uniforme, y en algunos casos nos indican que son sobreestimados. Este hecho se ve claramente en las gráficas ECDF de los modelos Poisson, donde el efecto de sobreestimar los p-valores de significación del valor se observa que es sistemático. Las simulaciones hechas a partir de la BN varian los resultados, en ocasiones se sobreestiman y en otros casos se infravalora.

En el caso concreto de la base de datos creados a partir de una $BN(\mu = 20, \lambda = 2)$, incorporando un 20 % de ceros vemos que se infraestiman los datos, pero si aumentamos los ceros hasta un 50 % observamos el efecto contrario, estas gráficas corresponden a la figura 5.1, reflejando un comportamiento parecido a los modelos hechos a partir de la distribución Poisson.

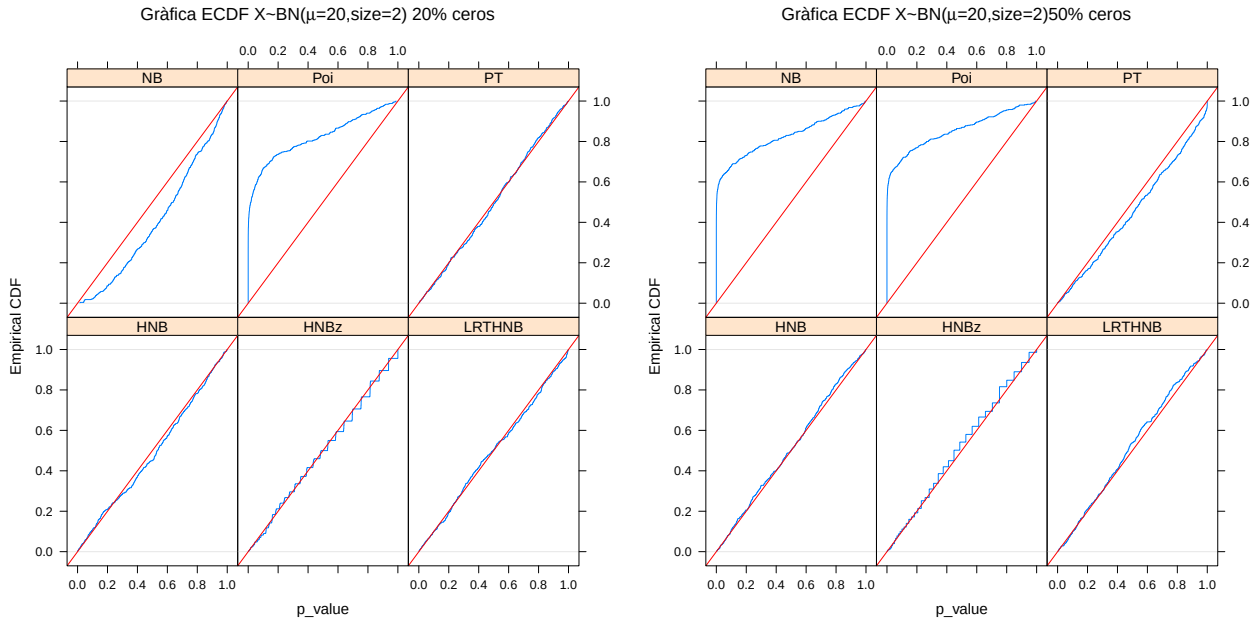


Figura 5.1: Gráficas ECDF (empirical cumulative distribution function) para datos creados a partir de una BN , con parámetro $\mu = 20$, $size = 2$ y 50 % de ceros. Como observamos en estas gráficas los modelos con un comportamiento más correcto son el PT , Hurdle y Cero Inflado, ya que los modelos basados en distribuciones Poisson y BN tienen un comportamiento que no se ajusta al que cabria esperar, que es que los p-valores se ajusten a la linea roja. La distribuciónes Poisson siempre sobreestima los datos y la BN tiene un comportamiento variado dependiendo de los datos.

También hemos creado una tablas en las cuales recogemos la proporción de p-valores que hubiesen sido significativos, y por lo tanto se hubiese hecho una estimación errónea, es decir p-valores menores de 0.05, este valor sería el esperado teniendo en cuenta la forma en la que han sido generados los datos, y corresponde al error de tipo I de los modelos. Estas tablas nos serviran para ver como actuan los diferentes modelos.

En la figura 5.1, y las tablas 5.1 y 5.2 podemos ver los resultados de las simulaciones para evaluar qué modelo estima de forma correcta el error de tipo I. Podemos observar como en la mayoria de los casos la distribución PT controla bien el error de tipo I. Los modelos que tienen en cuenta el exceso de ceros, como era de esperar, también presentan un buen comportamiento.

| | Poi | NB | PT | ZIPz | ZIP | LRTZIP | ZINBz | ZINB | LRTZINB | HPz | HP | LRTHP | HNBz | HNB | LRTHNB |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x$\sim P(2)$ $y$ 10 %$ceros$ | 0.08 | 0.05 | 0.05 | 0.02 | 0.04 | 0.05 | 0.02 | 0.04 | 0.05 | 0.06 | 0.04 | 0.05 | 0.06 | 0.04 | 0.05 |
| x$\sim P(2)$ $y$ 20 %$ceros$ | 0.12 | 0.06 | 0.05 | 0.05 | 0.06 | 0.07 | 0.05 | 0.06 | 0.06 | 0.05 | 0.06 | 0.07 | 0.05 | 0.06 | 0.06 |
| x$\sim P(2)$ $y$ 30 %$ceros$ | 0.11 | 0.03 | 0.04 | 0.04 | 0.04 | 0.04 | 0.04 | 0.04 | 0.03 | 0.04 | 0.04 | 0.04 | 0.04 | 0.04 | 0.03 |
| x$\sim P(2)$ $y$ 40 %$ceros$ | 0.13 | 0.03 | 0.05 | 0.03 | 0.06 | 0.05 | 0.03 | 0.06 | 0.05 | 0.04 | 0.06 | 0.05 | 0.04 | 0.06 | 0.05 |
| x$\sim P(2)$ $y$ 50 %$ceros$ | 0.17 | 0.04 | 0.00 | 0.04 | 0.05 | 0.06 | 0.04 | 0.04 | 0.06 | 0.07 | 0.05 | 0.06 | 0.07 | 0.04 | 0.06 |
| x$\sim P(20)$ $y$ 10 %$ceros$ | 0.25 | 0.09 | 0.02 | 0.06 | 0.06 | 0.06 | 0.06 | 0.06 | 0.06 | 0.06 | 0.06 | 0.06 | 0.06 | 0.06 | 0.07 |
| x$\sim P(20)$ $y$ 20 %$ceros$ | 0.44 | 0.44 | 0.04 | 0.05 | 0.05 | 0.05 | 0.05 | 0.04 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.04 | 0.05 |
| x$\sim P(20)$ $y$ 30 %$ceros$ | 0.46 | 0.46 | 0.27 | 0.04 | 0.06 | 0.04 | 0.04 | 0.05 | 0.04 | 0.04 | 0.06 | 0.04 | 0.04 | 0.05 | 0.04 |
| x$\sim P(20)$ $y$ 40 %$ceros$ | 0.52 | 0.52 | 0.09 | 0.06 | 0.04 | 0.05 | 0.06 | 0.04 | 0.06 | 0.06 | 0.04 | 0.05 | 0.06 | 0.04 | 0.05 |
| x$\sim P(20)$ $y$ 50 %$ceros$ | 0.52 | 0.52 | 0.02 | 0.05 | 0.06 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.06 | 0.05 | 0.05 | 0.05 | 0.05 |

Tabla 5.1: Tabla proporción p-valores menores de 0,05 (error tipo I empírico), datos generados con distribución Poisson y un exceso de ceros

En la tabla 5.2 presentamos los valores de la tabla obtenida para las simulaciones hechas con la distribución Binomial Negativa.

| | Poi | NB | PT | ZIPz | ZIP | LRTZIP | ZINBz | ZINB | LRTZINB | HPz | HP | LRTHP | HNBz | HNB | LRTHNB |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x$\sim Bn(2,1{,}5)$ $y$ 10 %$ceros$ | 0.23 | 0.06 | 0.06 | 0.10 | 0.19 | 0.18 | 0.01 | 0.04 | 0.08 | 0.08 | 0.19 | 0.18 | 0.08 | 0.04 | 0.08 |
| x$\sim Bn(2,1{,}5)$ $y$ 20 %$ceros$ | 0.25 | 0.04 | 0.05 | 0.03 | 0.16 | 0.10 | 0.01 | 0.04 | 0.03 | 0.04 | 0.16 | 0.10 | 0.04 | 0.04 | 0.03 |
| x$\sim Bn(2,1{,}5)$ $y$ 30 %$ceros$ | 0.29 | 0.05 | 0.06 | 0.06 | 0.18 | 0.15 | 0.03 | 0.06 | 0.06 | 0.06 | 0.18 | 0.15 | 0.06 | 0.06 | 0.06 |
| x$\sim Bn(2,1{,}5)$ $y$ 40 %$ceros$ | 0.24 | 0.03 | 0.05 | 0.04 | 0.14 | 0.11 | 0.03 | 0.03 | 0.03 | 0.04 | 0.14 | 0.11 | 0.04 | 0.03 | 0.03 |
| x$\sim Bn(2,1{,}5)$ $y$ 50 %$ceros$ | 0.28 | 0.02 | 0.01 | 0.04 | 0.12 | 0.09 | 0.03 | 0.04 | 0.04 | 0.04 | 0.12 | 0.09 | 0.04 | 0.04 | 0.04 |
| x$\sim Bn(2,2)$ $y$ 10 %$ceros$ | 0.18 | 0.04 | 0.04 | 0.05 | 0.11 | 0.10 | 0.00 | 0.04 | 0.04 | 0.04 | 0.11 | 0.10 | 0.04 | 0.04 | 0.04 |
| x$\sim Bn(2,2)$ $y$ 20 %$ceros$ | 0.21 | 0.03 | 0.04 | 0.05 | 0.11 | 0.10 | 0.02 | 0.05 | 0.03 | 0.04 | 0.11 | 0.10 | 0.04 | 0.05 | 0.03 |
| x$\sim Bn(2,2)$ $y$ 30 %$ceros$ | 0.25 | 0.03 | 0.06 | 0.06 | 0.11 | 0.09 | 0.03 | 0.04 | 0.05 | 0.06 | 0.11 | 0.09 | 0.06 | 0.04 | 0.05 |
| x$\sim Bn(2,2)$ $y$ 40 %$ceros$ | 0.25 | 0.03 | 0.03 | 0.06 | 0.13 | 0.12 | 0.05 | 0.06 | 0.04 | 0.06 | 0.13 | 0.12 | 0.06 | 0.06 | 0.04 |
| x$\sim Bn(2,2)$ $y$50 %$ceros$ | 0.24 | 0.04 | 0.00 | 0.06 | 0.17 | 0.13 | 0.05 | 0.04 | 0.06 | 0.06 | 0.17 | 0.13 | 0.06 | 0.04 | 0.06 |
| x$\sim Bn(20,1{,}5)$ $y$ 10 %$ceros$ | 0.64 | 0.03 | 0.05 | 0.05 | 0.60 | 0.55 | 0.04 | 0.06 | 0.06 | 0.05 | 0.60 | 0.55 | 0.05 | 0.06 | 0.06 |
| x$\sim Bn(20,1{,}5)$ $y$ 20 %$ceros$ | 0.63 | 0.02 | 0.05 | 0.05 | 0.59 | 0.56 | 0.05 | 0.05 | 0.05 | 0.05 | 0.59 | 0.56 | 0.05 | 0.05 | 0.05 |
| x$\sim Bn(20,1{,}5)$ $y$ 30 %$ceros$ | 0.64 | 0.00 | 0.04 | 0.05 | 0.55 | 0.51 | 0.04 | 0.06 | 0.04 | 0.05 | 0.55 | 0.51 | 0.05 | 0.06 | 0.04 |
| x$\sim Bn(20,1{,}5)$ $y$ 40 %$ceros$ | 0.68 | 0.01 | 0.04 | 0.06 | 0.62 | 0.58 | 0.05 | 0.05 | 0.05 | 0.06 | 0.62 | 0.58 | 0.06 | 0.05 | 0.05 |
| x$\sim Bn(20,1{,}5)$ $y$ 50 %$ceros$ | 0.73 | 0.15 | 0.06 | 0.05 | 0.61 | 0.58 | 0.06 | 0.06 | 0.06 | 0.05 | 0.61 | 0.58 | 0.05 | 0.06 | 0.06 |
| x$\sim Bn(20,2)$ $y$ 10 %$ceros$ | 0.56 | 0.02 | 0.05 | 0.03 | 0.53 | 0.49 | 0.03 | 0.04 | 0.04 | 0.03 | 0.53 | 0.49 | 0.03 | 0.04 | 0.04 |
| x$\sim Bn(20,2)$ $y$ 20 %$ceros$ | 0.59 | 0.02 | 0.05 | 0.05 | 0.52 | 0.46 | 0.05 | 0.06 | 0.05 | 0.05 | 0.52 | 0.46 | 0.05 | 0.06 | 0.05 |
| x$\sim Bn(20,2)$ $y$ 30 %$ceros$ | 0.65 | 0.01 | 0.06 | 0.06 | 0.55 | 0.50 | 0.06 | 0.05 | 0.05 | 0.06 | 0.55 | 0.50 | 0.06 | 0.05 | 0.05 |
| x$\sim Bn(20,2)$ $y$ 40 %$ceros$ | 0.66 | 0.28 | 0.05 | 0.05 | 0.57 | 0.53 | 0.05 | 0.04 | 0.05 | 0.05 | 0.57 | 0.53 | 0.05 | 0.04 | 0.05 |
| x$\sim Bn(20,2)$ $y$ 50 %$ceros$ | 0.67 | 0.64 | 0.05 | 0.04 | 0.55 | 0.52 | 0.05 | 0.05 | 0.04 | 0.04 | 0.55 | 0.52 | 0.04 | 0.05 | 0.04 |

Tabla 5.2: Proporción de p-valores menores de 0,05 (error tipo I empírico) , datos generados con distribución Binomial Negativa y un exceso de ceros

En segundo lugar hemos utilizado, para las simulaciones, otra serie de datos generados con R, y que siguen una distribución PT. La función utilizada para crear los datos es `rPT()`, y hemos generado bases con diferentes valores para los 3 parámetros que tiene la distribución.
Los parámetros que hemos utilizado son:$a = -0{,}5, -4, -100$, $\mu = 2, 4, 10$, $D = 2, 6, 20$, hemos realizados simulaciones para todos los escenarios con estos tres valores. Los resultados pueden verse en la Tabla 5.3.

Al simular datos segun una PT no es necesario simular una mixtura para los ceros,por que según que parámetros de la PT ya corresponden a distribuciones con una gran proporción de ceros.

De los resultados obtenidos de esta serie de simulaciones destacamos la gráfica de los datos simulados a partir de una PT con $a = -0{,}5$, $\mu = 2$ y el parámetro de dispersión igual a 20. En las simulaciones realizadas con estos parámetros, observamos que la distribución Poisson sobreestima los datos, como era de esperar, pero observamos

que tampoco la distribución BN ni la Hurdle BN estiman correctamente los datos. En (Figura 5.2) podemos ver las gráficas de distribución empíricas donde vemos que el modelo BN y Hurdle tienden a sobreestimar los datos, sobretodo prestamos mayor atención a la zona de significación (Figura 5.2 derecha) donde solo el modelo PT tiene un comportamiento correcto.

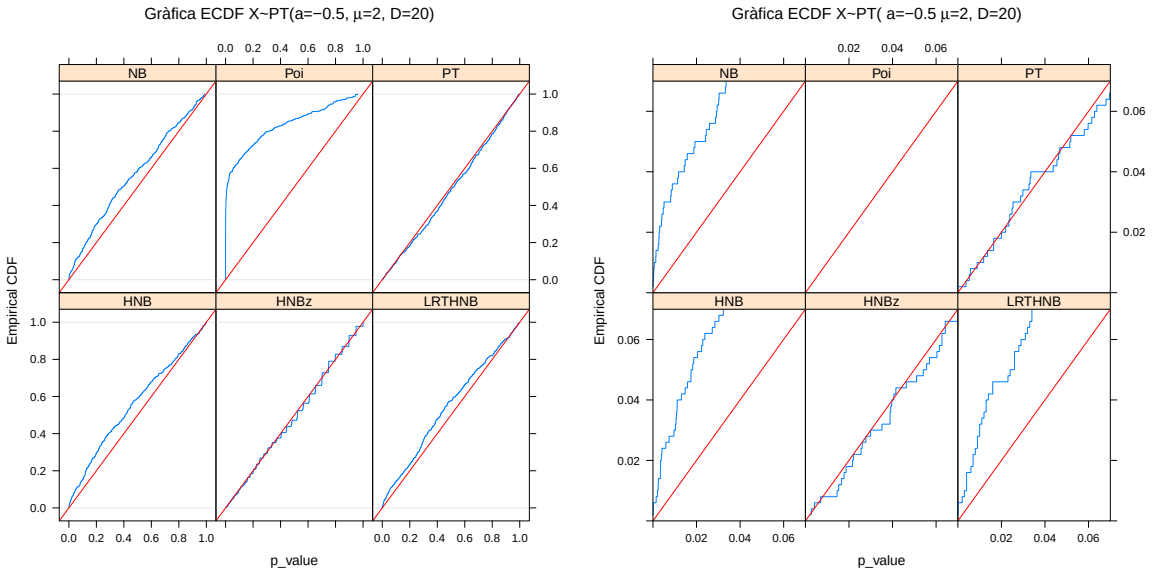


Figura 5.2: En estas dos gráficas, observamos que solo los modelos PT, estiman correctamente estas series de datos, en la segunda gráfica, hacemos un zoom para observar como se comportan en la zona crítica del intervalo (0,0.05). Observamos que en la zona de significación los modelos PT estiman bien los datos

En la Figura 5.3 también observamos como en la simulación hecha con los parámetros $a = -4$, $\mu = 2$ y el parámetro de dispersión 20, los modelos PT tienen un mejor comprtamiento

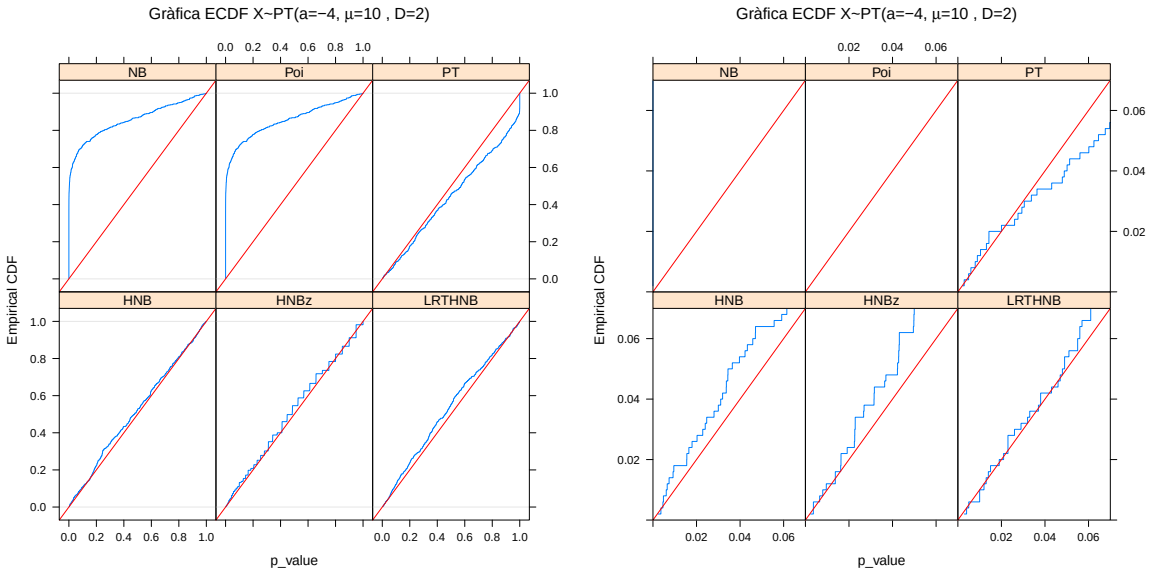


Figura 5.3: En estas dos gráficas, observamos que los modelos PT, estiman correctamente estas series de datos, en la segunda gráfica, hacemos un zoom para observar como se comportan en la zona crítica del intervalo (0,0.05). Observamos que en la zona de significación los modelos PT estiman bien los datos

La tabla 5.3 es la tabla resultante de las simulaciones correspondientes a los datos generados según una PT, donde vemos que los modelos PT presentan major estabilidad en el error empírico de tipo I.

| | Poi | NB | PT | ZIPz | ZIP | LRTZIP | ZINBz | ZINB | LRTZINB | HPz | HP | LRTHP | HNBz | HNB | LRTHNB |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x$\sim PT(a = 0{,}5 mu = 2 D = 2)$ | 0.15 | 0.05 | 0.05 | 0.05 | 0.13 | 0.12 | 0.00 | 0.05 | 0.03 | 0.04 | 0.13 | 0.12 | 0.04 | 0.05 | 0.05 |
| x$\sim PT(a = 0{,}5 mu = 2 D = 6)$ | 0.40 | 0.06 | 0.06 | 0.06 | 0.36 | 0.31 | 0.00 | 0.07 | 0.03 | 0.05 | 0.36 | 0.31 | 0.05 | 0.05 | 0.06 |
| x$\sim (a = 0{,}5 mu = 2 D = 20)$ | 0.59 | 0.11 | 0.05 | 0.05 | 0.57 | 0.52 | 0.00 | 0.11 | 0.04 | 0.05 | 0.57 | 0.52 | 0.05 | 0.10 | 0.09 |
| x$\sim (a = 0{,}5 mu = 5 D = 6)$ | 0.43 | 0.09 | 0.06 | 0.04 | 0.38 | 0.34 | 0.00 | 0.09 | 0.03 | 0.05 | 0.38 | 0.34 | 0.05 | 0.08 | 0.06 |
| x$\sim (a = 0{,}5 mu = 5 D = 20)$ | 0.42 | 0.08 | 0.06 | 0.04 | 0.39 | 0.35 | 0.00 | 0.08 | 0.04 | 0.04 | 0.39 | 0.35 | 0.04 | 0.08 | 0.05 |
| x$\sim (a = 0{,}5 mu = 10 D = 10)$ | 0.54 | 0.09 | 0.06 | 0.04 | 0.53 | 0.48 | 0.00 | 0.09 | 0.04 | 0.04 | 0.53 | 0.48 | 0.04 | 0.08 | 0.08 |
| x$\sim (a = 0{,}5 mu = 10 D = 20)$ | 0.67 | 0.13 | 0.06 | 0.03 | 0.65 | 0.60 | 0.00 | 0.13 | 0.06 | 0.03 | 0.65 | 0.60 | 0.03 | 0.10 | 0.08 |
| x$\sim (a = -4 mu = 5 D = 2)$ | 0.16 | 0.06 | 0.07 | 0.02 | 0.15 | 0.11 | 0.00 | 0.07 | 0.14 | 0.03 | 0.15 | 0.11 | 0.03 | 0.07 | 0.05 |
| x$\sim (a = -4 mu = 2 D = 2)$ | 0.15 | 0.04 | 0.05 | 0.04 | 0.10 | 0.09 | 0.00 | 0.05 | 0.04 | 0.04 | 0.10 | 0.09 | 0.04 | 0.05 | 0.04 |
| x$\sim (a = -4 mu = 2 D = 6)$ | 0.44 | 0.01 | 0.05 | 0.04 | 0.22 | 0.18 | 0.04 | 0.06 | 0.05 | 0.05 | 0.22 | 0.18 | 0.05 | 0.06 | 0.05 |
| x$\sim PT(a = -4 mu = 2 D = 20)$ | 0.67 | 0.67 | 0.04 | 0.06 | 0.41 | 0.37 | 0.06 | 0.05 | 0.05 | 0.06 | 0.41 | 0.37 | 0.06 | 0.04 | 0.34 |
| x$\sim PT(a = -4 mu = 5 D = 2)$ | 0.18 | 0.05 | 0.05 | 0.04 | 0.17 | 0.15 | 0.00 | 0.07 | 0.12 | 0.04 | 0.17 | 0.15 | 0.04 | 0.07 | 0.05 |
| x$\sim PT(a = -4 mu = 5 D = 6)$ | 0.43 | 0.02 | 0.05 | 0.04 | 0.32 | 0.26 | 0.04 | 0.05 | 0.04 | 0.04 | 0.32 | 0.26 | 0.04 | 0.05 | 0.04 |
| x$\sim PT(a = -4 mu = 5 D = 20)$ | 0.65 | 0.65 | 0.03 | 0.05 | 0.47 | 0.41 | 0.05 | 0.05 | 0.05 | 0.05 | 0.47 | 0.41 | 0.05 | 0.05 | 0.05 |
| x$\sim PT(a = -4 mu = 10 D = 6)$ | 0.42 | 0.02 | 0.05 | 0.05 | 0.37 | 0.33 | 0.05 | 0.05 | 0.05 | 0.05 | 0.37 | 0.33 | 0.05 | 0.05 | 0.05 |
| x$\sim PT(a = -4 mu = 10 D = 20)$ | 0.66 | 0.66 | 0.04 | 0.07 | 0.53 | 0.51 | 0.07 | 0.06 | 0.05 | 0.07 | 0.53 | 0.51 | 0.07 | 0.06 | 0.05 |
| x$\sim PT(a = -100 mu = 2 D = 2)$ | 0.19 | 0.04 | 0.06 | 0.06 | 0.11 | 0.10 | 0.02 | 0.04 | 0.06 | 0.07 | 0.11 | 0.10 | 0.07 | 0.04 | 0.06 |
| x$\sim PT(a = -100 mu = 2 D = 6)$ | 0.41 | 0.06 | 0.06 | 0.05 | 0.12 | 0.10 | 0.05 | 0.04 | 0.04 | 0.05 | 0.12 | 0.10 | 0.05 | 0.04 | 0.04 |
| x$\sim PT(a = -100 mu = 2 D = 20)$ | 0.66 | 0.66 | 0.00 | 0.04 | 0.20 | 0.16 | 0.04 | 0.07 | 0.15 | 0.04 | 0.20 | 0.16 | 0.04 | 0.08 | 0.40 |
| x$\sim PT(a = -100 mu = 5 D = 2)$ | 0.18 | 0.05 | 0.05 | 0.03 | 0.14 | 0.13 | 0.00 | 0.05 | 0.10 | 0.04 | 0.14 | 0.13 | 0.04 | 0.06 | 0.06 |
| x$\sim PT(a = -100 mu = 5 D = 20)$ | 0.65 | 0.65 | 0.00 | 0.05 | 0.27 | 0.23 | 0.05 | 0.07 | 0.05 | 0.05 | 0.27 | 0.23 | 0.05 | 0.07 | 0.07 |
| x$\sim PT(a = -100 mu = 10 D = 2)$ | 0.31 | 0.14 | 0.13 | 0.00 | 0.57 | 0.71 | 0.00 | 0.29 | 0.43 | 0.00 | 0.57 | 0.71 | 0.00 | 0.29 | 0.43 |
| x$\sim PT(a = -100 mu = 10 D = 6)$ | 0.43 | 0.01 | 0.06 | 0.06 | 0.39 | 0.32 | 0.06 | 0.05 | 0.05 | 0.06 | 0.39 | 0.32 | 0.06 | 0.05 | 0.05 |
| x$\sim PT(a = -100 mu = 10 D = 20)$ | 0.68 | 0.68 | 0.00 | 0.04 | 0.38 | 0.34 | 0.04 | 0.07 | 0.05 | 0.04 | 0.38 | 0.34 | 0.04 | 0.07 | 0.05 |

Tabla 5.3: Tabla proporción p-valores menores de 0,05 (error tipo I empírico) con datos aleatorios con distribución PT. En esta tabla podemos observar que la estabilidad de los modelos PT con datos generados a partir de una PT de diferentes parámetros es clara en comparación con los modelos clásicos.

### 5.1.2. Comparativas a partir de los AICs

Otro estadístico que usamos para la comparación de los modelos son los AIC, o Criterio de información de Akaike. Este estadístico es la medida de la calidad relativa de un modelo estadístico para un conjunto de datos, y se calcula de mediante la fórmula:

$$AIC = 2k - 2 * ln(L)$$

donde $k$ corresponde al número de parámetros en el modelo estadístico y $L$ es el máximo valor de la función de probabilidad para el modelo estimado.

El criterio de selección es, escoger modelos con valores más bajos de AIC que corresponde al modelo que mejor explica los datos con el mínimo número de parámetros es el que presenta más bajo el valor de AIC. La lógica que sigue este método no es la de las pruebas de hipótesis. Por tanto, no se debe plantear una hipótesis nula o calcular un p-valor , y no es necesario decidir acerca de la tendencia del valor para determinar su significación estadística. Además, el método permite determinar qué modelo es el más parecido al correcto y cuantificar su parecido. Para comparar los AIC de los diferentes modelos los representamos usando diagramas de cajas donde hemos representado con un punto rojo la media de los AICs de los modelos simulados.

Destacamos los resultados de la Figura 5.4 donde vemos los resultados de la simulación hecha a partir de una

BN($\mu$ = 20,size=2) y incorporando un 20 % de ceros.

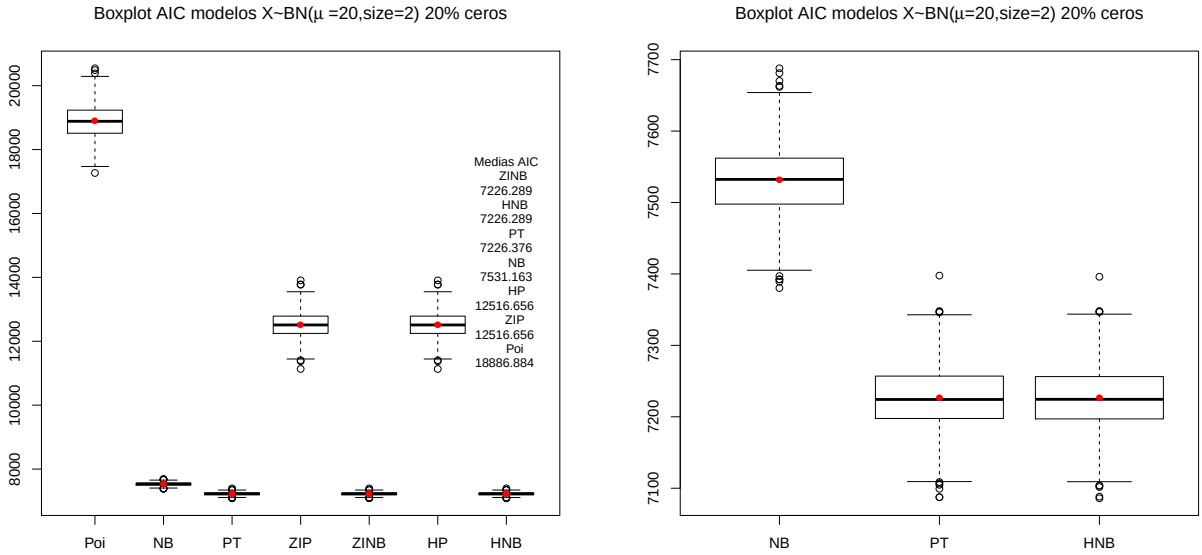


Figura 5.4: Diagramas de caja para los AIC de los modelos simulados a partir de una BN($\mu$ = 20,size=2), incorporando un 20 % de ceros. Observamos entre los modelos Hurdle(BN), Cero inflado (BN), y PT hay unas diferencias mínimas.

De los resultados obtenidos con datos simulados a partir de una PT también observamos que los AICs de los modelos PT también son menores que los de los demas modelos. En los la figura 5.5 observamos los diagramas de cajas de una serie de datos generados a partir de una PT con a=-0.5, $\mu$ = 2 y el parámetro de dispersión igual a 20. En esta simulación hemos comprobado cada en cada una de las series generadas y el menor AIC lo presenta siempre el modelo PT.

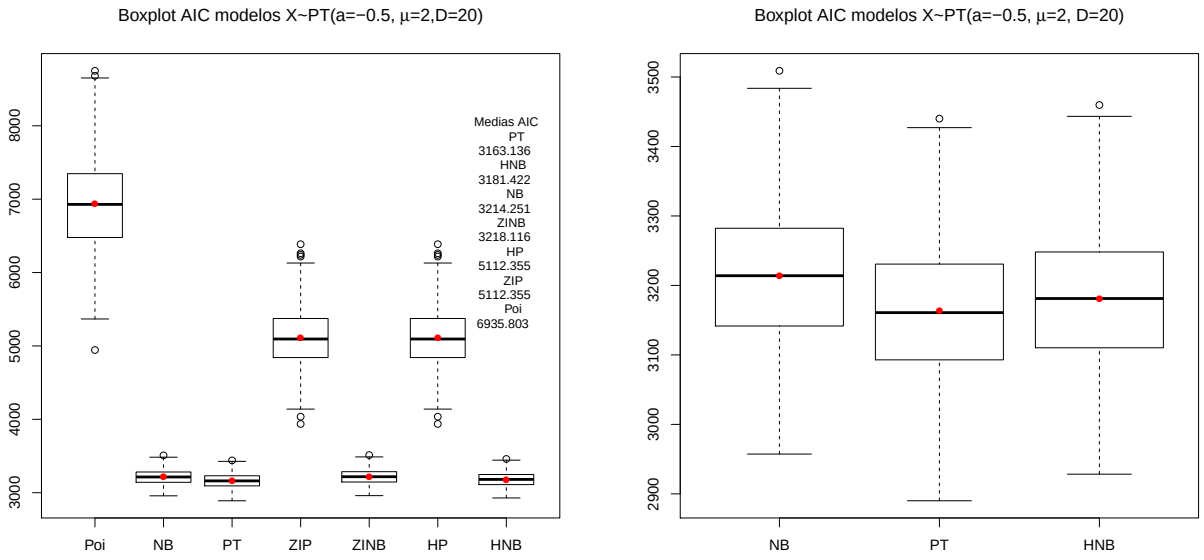


Figura 5.5: Diagramas de caja para los AIC de los modelos simulados partir de una PT con a=-0.5, $\mu = 2$ y el parámetro de dispersión igual a 20. Observamos que los modelos PT presentan los menores AICs

## 5.2. Simulaciones con datos reales

En esta sección presentamos las simulaciones con datos reales. Para realizar estas simulaciones, decidimos permutar la covariable (variable explicativa), para demostrar que el mejor comportamiento de un modelo no dependa del modelo con el que se haya simulado los datos. En particular hemos utilizado los datos genéticos de los nigerianos descritos en la sección 2, utilizando como covariable la variable sexo (hombre, mujer) y analizando los datos con cada modelo, obteniendo los mismos estadísticos que en las anteriores simulaciones y realizando la misma serie de gráficas.

### 5.2.1. Error de tipo I

Para las simulaciones, tenemos, la variable respuesta que son los conteos de cada gen en los diferentes individuos y también tenemos el factor para permutar, que en este caso sera el sexo. Por lo tanto escojemos una serie de datos del conteo de un gen para los diferentes individuos, y realizamos los modelos permutando el factor que en este caso es la covariable que nos indica el sexo. Una permutacion aleatoria de la covariable nos producirá el efecto deseado, que es, que esa covariable sea independiente a los datos, por tanto los p-valores de los modelos si generamos una muestra lo suficientemente grande debería de distribuirse uniformemente en el intervalo (0,1).

Con este tipo de datos nos encontramos que obtenemos resultados variados, pero observamos que hay una serie de datos en los cuales por sistema los modelos clásicos sobrestiman los datos en cambio los modelos creados a partir de la PT los resultados son los correctos.

En las gráficas de la Figura 5.6 se observa con claridad. Vemos que en el gen 11 los modelos clásicos no estiman bien los datos, ya que el error de tipo I empírico no se distribuye correctamente, también sucede esto con el gen 14, en cambio vemos como en las simulaciones hechas con el gen 2 el modelo BN también es adecuado.

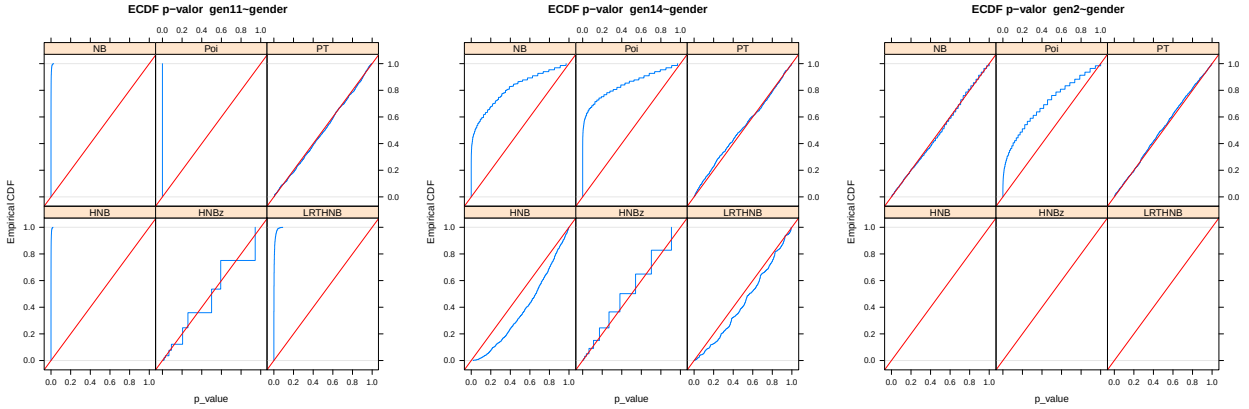


Figura 5.6: En estas tres gráficas, observamos que para las simulaciones generadas a partir de los diferentes genes observamos diferentes comportamientos, destacamos los resultados del gen que hemos numerado con el número 11, en este gen vemos que todos los modelos escepto los modelos PT fallan de manera clara, en el gen número 2 en cambio vemos que los modelos BN también estiman bien los datos, el gen numero 14 los modelos PT también son los que mejor funcionan

### 5.2.2. Comparativas a partir de los AICs

Destacamos los resultados que podemos ver en la Figura 5.7 donde hemos realizado los diagramas de cajas de los AIC de los modelos obtenidos de las simulaciones hechas con los datos reales, donde podemos ver que los modelos PT obtienen en promedio unos valores del estadístico AIC menores, lo que nos indica una mayor eficiencia de los modelos PT.

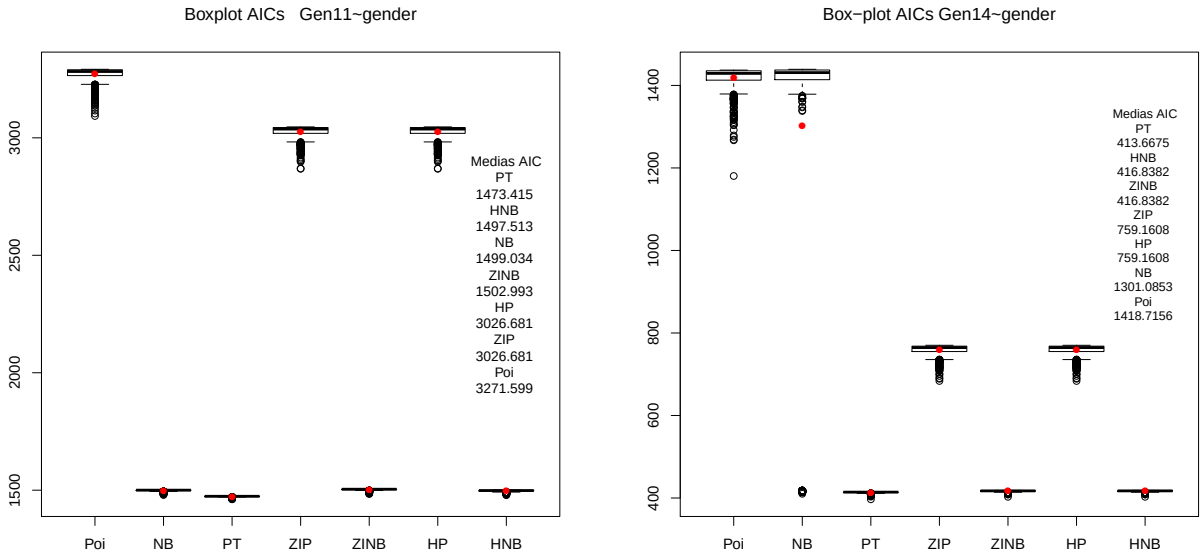


Figura 5.7: Observamos en estas gráficas que los AICs menores pertenecen a los modelos PT

## 5.3. Estudio de potencia

Con el objetivo de comparar la sensibilidad o potencia (error de tipo II) de los diferentes modelos, realizamos una serie de simulaciones. La potencia de un modelo probabilistico es la capacidad de un modelo para detectar diferencias cuando estas son reales.

Para tal efecto creamos la función `simulapower()`, que adjuntamos en el anexo, dicha función, crea dos series de datos con diferentes medias, y un factor que nos indica a que grupo pertenece cada dato a partir de aquí, modelamos los datos en función del grupo al que pertenece.

Esta operación la realizamos 500, 1000 y 2000 veces y obtenemos la proporción de p-valores que resultan significativos, es decir los p-valores que nos dicen que hay diferencias cuando esa diferencia es real, a continuación aumentamos en una unidad unos de los grupos y repetimos los modelos, obteniendo de nuevo los estadísticos. Estas proporciones de p-valores significativos serán los que luego representaremos en las gráficas.

Los datos los creamos a partir de la función `rPT()`. Realizamos las simulaciones para datos creados con diferentes parámetros de dispersión y forma. Este tipo de simulaciones las realizamos con diferente número de observaciones y con diferentes medias, de esta manera intentamos observar la capacidad de los modelos de reconocer que los grupos son diferentes en el caso que realmente lo sean.

En estas simulaciones los modelos Hurdle y Cero inflado no los tomamos en consideración por que solo con medias muy bajas obteniamos algun cero (condición esencial para poder utilizar ese tipo de modelos), asi que para comparar la potencia de los modelos PT, lo comparamos con el modelo Binomial negativo, que es el que siempre podemos usar aunque no hayan ceros, pero que los datos sí presenten sobredispersión, de hecho es un modelo que ajusta bastante bien a gran cantidad de datos que presentan sobredispersión.

Prácticamente todos los casos estudiados observamos que los modelos PT tienen una capacidad mayor de diferenciar dos grupos que realmente son diferentes, que los modelos BN, y solo en casos con una $''n''$ muy pequeña y pequeñas diferencias los modelos BN son similares a los modelos PT.

La Figura 5.8 muestra que los modelos PT tienen más potencia para todos los escenarios simulados.

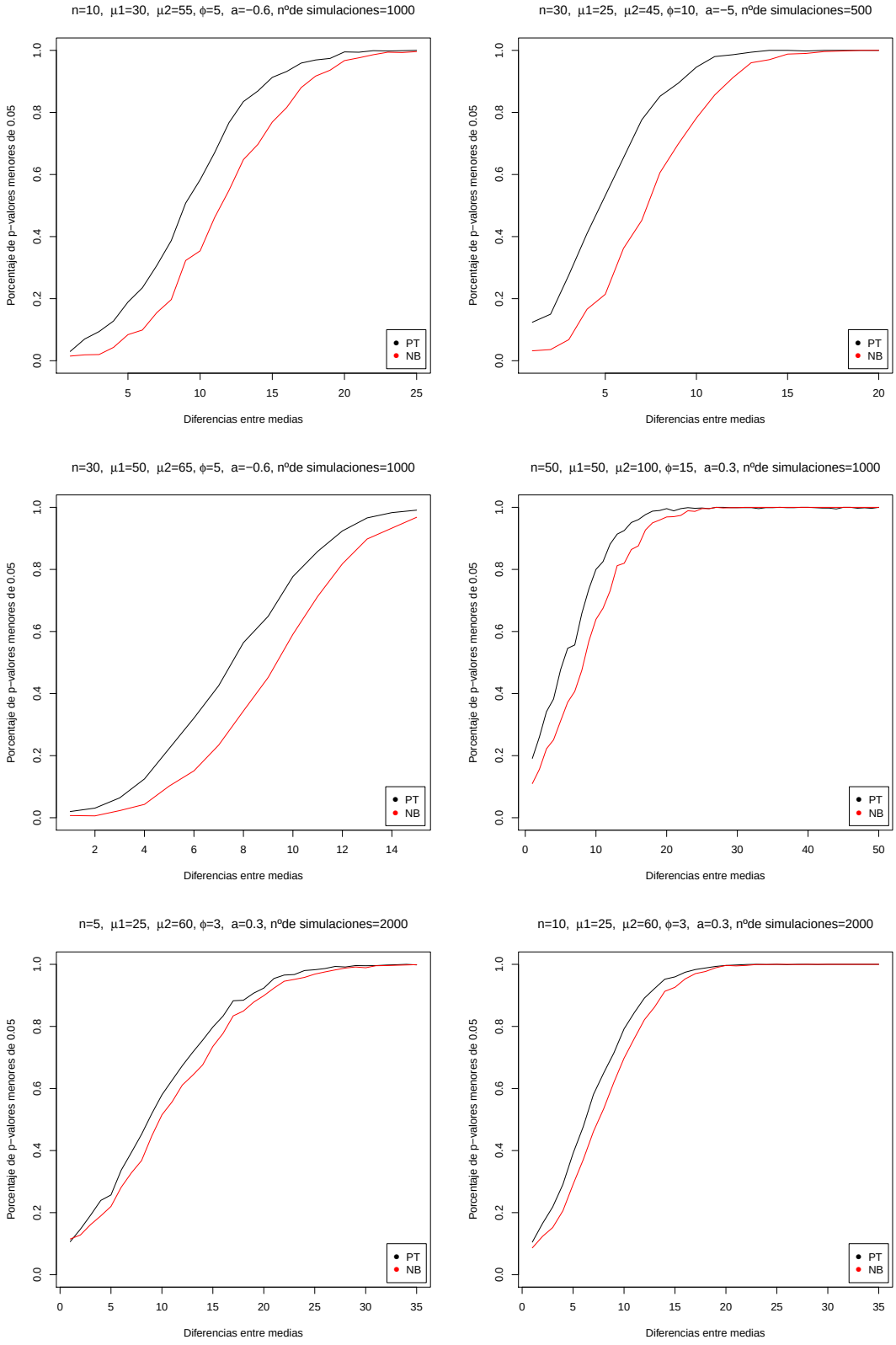


Figura 5.8: Curvas de potencia. Observamos que los modelos PT tienen una potencia mayor para discriminar si hay diferencia entre dos grupos que realmente son diferentes. En estas simulaciones el valor de la 'n', corresponde al número de de datos que forman cada uno de los grupos que hemos simulado, la $\mu 1$ corresponde al valor de la media inicial, y el valor de la $\mu 2$ corresponde al valor de la media del segundo grupo en la última simulación.

# Sección 6

# Aplicación a datos reales, uso de los modelos

Con tal de ilustrar el comportamiento de todos los modelos presentados en este trabajo, usaremos los datos de *Pickrell* [5] descritos en la sección 2. Nuestro objetivo es decidir si los genes estudiados se expresan de manera diferente en hombres y mujeres.

Para los genes estudiados disponemos de la información biológica sobre si son o no específicos de género, es decir si la expresión génica debería ser distinta entre hombres y mujeres.

Analizamos 15 genes utilizando los distintos modelos que hemos descrito en la sección 2, los resultados pueden verse en la tabla 6.1. A parte hemos estimado el parámetro de forma de la PT, y el p-valor del test de hipótesis realizado bajo la hipótesis nula de que los datos siguen una Binomial negativa ($H_0 : a = 0$). Con estos dos valores valoraremos cual de los modelos es el más adecuado.

| | Poisson | Binomial neg | Hurdle BN Conteo | Hurdle BN ceros | Poisson-Tweedie | $a$ | p-valor BN | Distibución |
|---|---|---|---|---|---|---|---|---|
| Gen 1 | 0.684 | 0.776 | 0.996 | 0.396 | 0.606 | -0.81 | 0.76 | Bin. Neg. |
| Gen 2 | 0.250 | 0.564 | | | 0.331 | 0.17 | 0.82 | Bin. Neg. |
| Gen 3 | 8.15e-30 | 0.127 | | | 0.949 | -1.39 | NA | Bin. Neg. |
| Gen 4 | 0.308 | 0.931 | | | 0.427 | -0.71 | 0.088 | Bin. Neg. |
| Gen 5 | 5.11e-09 | 0.414 | | | 0.300 | -0.21 | 0.78 | Bi.n Neg. |
| Gen 6 | 0.444 | 0.551 | 0.066 | 0.298 | 0.751 | -135.55 | 0.28 | Bin. Neg. |
| Gen 7 | 0 | 0.174 | | | 0.496 | 0 | 1 | Bin. Neg. |
| Gen 8 | 0.014 | 0.442 | | | 8.18e-01 | -16.12 | 0.018 | Neyman Type A |
| Gen 9 | 4.59e-32 | 0.044 | | | 0.982 | -0.96 | 0.52 | Bin. Neg. |
| Gen 10 | 0.907 | 0.986 | | | 0.504 | -7.81 | 0.055 | Bin. Neg. |
| Gen 11 | 0.37e-39 | 2.11e-05 | 2e-16 | 0.06 | 0.972 | 0.7 | 6.1 e-14 | Poi. Inv. Gaussiana Generalizada |
| Gen 12 | 4.68e-108 | 2.00e-101 | 2e-16 | 0.99 | 1.10e-59 | -0.5 | 5.7e-5 | Polya-Aeppli |
| Gen 13 | 0.993 | 0.995 | | | 8.59e-06 | -3.95 | 0.32 | Bin. Neg. |
| Gen 14 | 3.32e-63 | 5.64e-55 | 1.61e-08 | 0.995 | 1.29e-12 | -0.55 | 0.018 | Polya-Aeppli |
| Gen 15 | 9.89e-21 | 2.87e-20 | 8.03e-08 | 0.997 | 1.29e-12 | -3.52 | 7.1e-11 | Neyman Type A |

Tabla 6.1: en esta tabla podemos ver en las columnas los p-valores de significacion del sexo como variable explicativa de los modelos, Poisson, Binomial Negativo, Hurdle Binomial Negativo y PT. También hemos incorporado el valor del parámetro $a$, que corresponde al parámetro de forma de la distribución PT y el p-valor de que la distribución de los datos siga una Binomial negativa. Este valor es el resultante de un contraste de hipótesis bajo la hipótesis nula de que los datos siguen una distribución BN.

Observamos que hay series de datos que tienen resultados contradictorios, como por ejemplo el gen 11. Los análisis de este gen, utilizando los modelos Poisson, Binomial Negativo y Hurdle, indican que la expresión génica es

distinta entre hombres y mujeres. Por otro lado el modelo PT muestra que esta diferencia no es estadísticamente significativa. Sabemos que este gen, corresponde al gen de la secretina(SCT), que codifica un neuropéptido de la familia del VIP/glucagón, necesario para el correcto desarrollo del cerebro, y este gen no debería de expresarse de manera diferente entre hombres y mujeres. Es por ello que los resultados obtenidos con los modelos Poisson, Binomial Negativo y Hurdle son falsos positivos.

La figura 6.1 muestramos el ajuste de la distribución Poisson, Binomial Negativa y PT a los datos. Observamos como la PT se ajusta perfectamente en hombres y mujeres y como BN y P no lo hacen bien en mujeres. Esto es debido a que hay un individuo con un conteo muy grande que hace la distribución presente una cola pesada que las distribuciones Poisson y BN no modelan bien. Notemos que el test para el parámetro de forma de la PT rechaza la hipótesis que una Binomial Negativa sea adecuada ($H_0 : a = 0$). Esto corrobora también el hecho de que la BN no es un buen modelo para este gen. Para el análisis del gen 11 sería, pues, más adequado el uso de los modelos basados en la distribución PT, ya que con los modelos Poisson y Binomial Negativo obtenemos resultados erróneos.

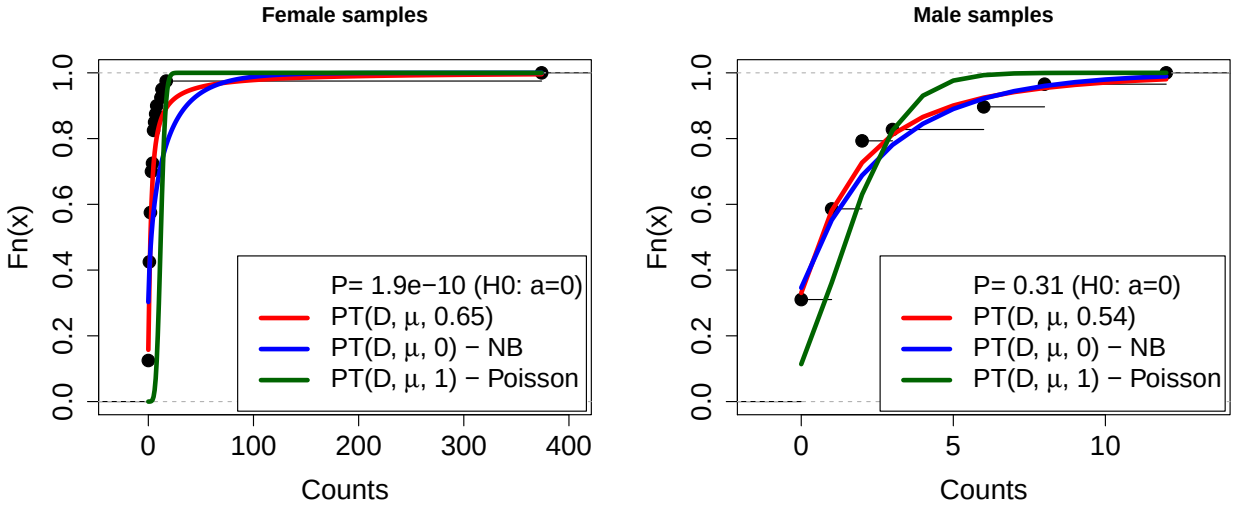


Figura 6.1: En estas gráficas vemos representado, con puntos negros, la función empírica de distribución, la linea verde representa la distribución Poisson, con los parámetros estimados, la linea azul, la Binomial Negativa y la linea roja la PT. En este ejemplo vemos como en el caso de las mujeres, la presencia de un dato muy por encima de la media hace que el ajuste de las distribuciones P y NB, no sea bueno, las distribución PT en cambio recoge mejor la distribución de los datos a pesar de tener un dato que podriamos pensar que es anómalo. En la gráfica de los hombres vemos que también la PT se comporta mejor, aunque la distribución BN también es fáctible. Los p-valores se refieren a la probabilidad que los datos sigan una distribución Binomial Negativa ($a = 0$), vemos que en el caso de las mujeres este es muy pequeño lo que indica el mal ajuste de la distribución BN.

Por otro lado el gen 13 tiene un comportamiento al contrario, sale significativo con el modelo PT y no con Poisson y Binomial Negativa. Este gen es un gen específico del hombre de la región del cromosoma Y que debería mostrar diferencias estadísticas significativas en la expresión génica entre hombres y mujeres, tal y como hace la PT y no la Poisson y Binomial Negativa. Este ejemplo nos ilustra el hecho de que los modelos PT tienen mayor capacidad de detectar las diferencias entre dos grupos, comportamiento que ya hemos observado en las simulaciones.

Con tal de ver el mal ajuste que pueden presentar los datos reales a las distribuciones clásicas destacamos la

gráfica de la figura 6.2, hecha a partir de los datos del gen 15, donde se observa que la distribuciones clásicas se alejan bastante de la verdadera distribución de la variable.

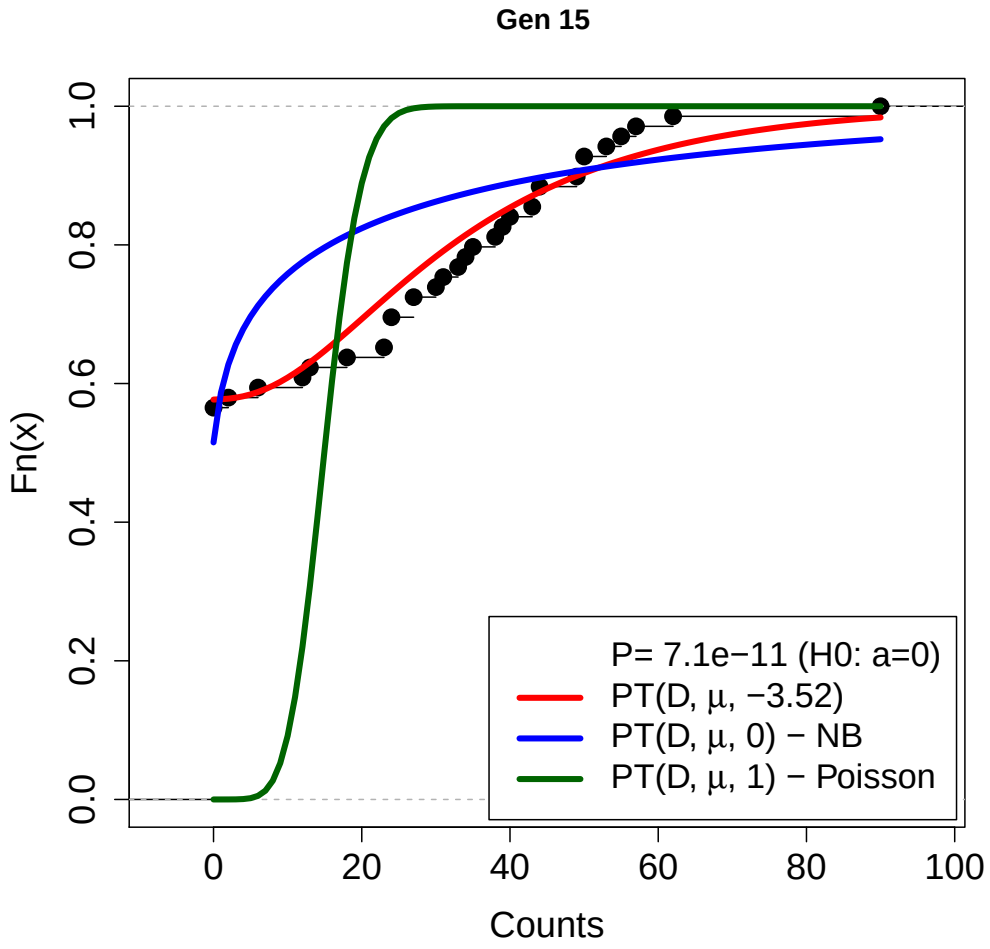


Figura 6.2: En esta gráfica comparamos las distribuciones empíricas y las estimadas dels gen etiquetado en la matriz original como "ENSG00000099749", que corresponde a nuestro gen 15. Observamos, mediante este ejemplo que las distribuciones clásicas pueden alejar mucho de las distribuciones empíricas, en cambio la PT ajusta mejor.

# Sección 7

# Conclusiones finales

Como resultado de las simulaciones hechas y de los resultados obtenidos de la aplicación del modelo en datos reales, podemos decir que la modelización de conteos mediante la distribución PT presenta bastantes ventajas respecto a los modelos Poisson, Binomial Negativo (incluyendo sus versiones cero infladas y Hurdle). Las ventajas del uso de modelo PT, que hemos encontrado son:

1. Modelo muy flexible para datos de conteos, ya que es una generalización que incluye diversas distribuciones usadas para en datos de conteos dependiendo del valor del parámetro $a$ (Polya-Aeppli, Binomial Negativa, Poisson Inversa Gaussiana, Poisson).

2. Esta flexibilidad aporta mejor ajuste en datos reales que presentan características diversas, lo que resulta muy útil en genética que tiene que analizar miles de genes.

3. Las simulaciones realizadas nos han mostrado que los modelos PT muestran igual o mejores resultados que los modelos clásicos, cuando los datos son simulados usando estas distribuciones tradicionales. Cuando los datos se simulan según una PT, los resultados , como es de esperar, son mejores con el modelo PT. Para evitar el problema que siempre hay con los estudios de simulación, hemos usado datos reales en los que, mediante permutación, hemos generado datos que se podrian obtener en los estudios de RNA-seq que estamos interesados. Los resultados obtenidos nos han mostrado que, la PT claramente mejora el resto de modelos tanto en error de tipo I (falsos positivos), cómo en la potencia del modelo (capacidad de diferenciar dos poblaciones realmente diferentes).

4. En el análisis de datos reales, hemos comprobado el mejor comportamiento del modelo PT tanto en el control del error de tipo I (caso del gen 11), como en la potencia (caso del gen 13). Estos resultados los hemos podido obtener ya que para comparar la expresión génica entre hombres y mujeres, disponemos de una lista de genes que sabemos si deben mostrar diferencia de expresión en cuanto al género o no.

Una limitación que podria tener en el modelo basado en la distribución PT, es la no convergencia aunque esto hemos estimado que ha sucedido en menos del 1 % de los casos, y ha ocurrido en casos muy particulares. Esto es debido a que la estimación del modelo se realiza con medios númericos que no se pueden calcular. Este hecho nos ha provocado que algunas de las simulaciones realizadas no se han podido utilizar para ilustrar este trabajo.

En resumen podriamos decir que el modelo PT es recomendado para datos de ultrasecuenciación, obtenidos a través de la técnica RNA-seq.

# Sección 8

# Trabajo Futuro

En un trabajo futuro se debería de estudiarse el comportamiento del modelo PT incorporando más covariables de ajuste, ya que estos modelos se van a usar en el estudio de enfermedades complejas en el que otras variables confusoras pueden jugar un papel en el cambio de expresión génica.

En cuanto a las mejoras de este trabajo, se podrían optimizar las funciones con las cuales hemos realizado las simulaciones, ya que los bucles hechos con la funcion ”for”, se podrian substituir por funciones ”apply”, ”do.call”, que podrian acortar los tiempos de compilación.

# Sección 9

# Agradecimientos

A la Dr. Mercè Farré por sus comentarios y cambios realizados en una versión final del manuscrito.

# Bibliografía


[1] El-Shaarawi, A., Zhu, R., and Joe, H. (2011) Modelling species abundance using the Poisson-Tweedie family, *Environmetrics, 22:152-164*

[2] Aarón Salinas-Rodríguez, M en C; Betty Manrique-Espinoza, M en C; Sandra G Sosa-Rubí, Dr en C Análisis estadístico para datos de conteo: aplicaciones para el uso de los servicios de salud, *http://www.scielo.org.mx/scielo.php?script=sci$_a$rttextpid = S*0036 − 36342009000500007

[3] Antonio Alonso Alonso, 2011 ADN forense, investigación criminal y búsqueda de desaparecidos, *http://www.sebbm.es/ES/divulgacion-ciencia-para-todos_10/adn-forense–investigacion-criminal-y-busqueda-de-desaparecidos*$_6$04

[4] Achim Zeileis, Christian Kleiber, Simon Jackman Regression Models for Count Data in R, *http://cran.r-project.org/web/packages/pscl/vignettes/countreg.pdf*

[5] Pickrell J, Marioni J, Pai A, Degner J, Engelhardt B, Nkadori E, Veyrieras J, Stephens M, Gilad Y, Pritchard Understanding mechanisms underlying human gene expression variation with RNAsequencing, *Nature 2010, 464:768-772.*

[6] Wikipedia Secuenciación ADN, *http://es.wikipedia.org/wiki/Secuenciación_de_ADN*

[7] Gerber HU 1991 From the generalized gamma to the generalitzed negative binomial distribution, *Insurance: Mathematics and economics 10 : 303-309*

[8] Johnson NL, Kemp AW ,Kontz S 2005 Univariate discrete Distributions, 3rd edn , *Wiley Intersciencie: New York*

[9] Kokonendji CC, Dossou-Gbete S, Demetrico CGB. 2004, Some Discrete exponencial dispersión models: Poisson-Tweedie and Hinde-Demetrico classes, Statistics and Operations Research Transactions-SORT 28:201-2014.

[10] Chu Y, Corey DR, August 2012 RNA sequencing: platform selection, experimental design,and data interpretation, NA

[11] Hougaard P. Lee M LT, Whitmore GA 1997 Analisis of overdispersed count data by mixtures of Poisson variables and poisson processed, Biometrics 53:1225-1238

# Apéndice A

# Anexo

Las funciones que presentamos a continuación se pueden mejorar la programación en R ,en un futuro trabajo, evitando los bucles y sustituyendo estos por las funciones `apply()`, `do.call()`.

La función *hacerbase* nos da como resultado una base de datos con el número deseado tanto de filas como de columnas, también escojemos la distribucion y el numero de ceros que queremos para simular los datos con exceso de ceros.

```
> hacerbase<-function(numfil,numcol,numCeros,distribucion=NULL)
+ {
+ set.seed(123)
+ base<-matrix(rep(0,numfil*numcol),numfil,numcol)
+ for (i in 1:numfil)
+ {
+ if (distribucion==1)
+ {base[i,]<-sample(c((rep(0,numCeros)),rpois(numcol-numCeros,50)))}
+ if (distribucion==2)
+ {base[i,]<-sample(c((rep(0,numCeros)),rpois(numcol-numCeros,20)))}
+ if (distribucion==3)
+ {base[i,]<-sample(c((rep(0,numCeros)),rnbinom(numcol-numCeros, mu=2,size=1.5)))}
+ if (distribucion==4)
+ {base[i,]<-sample(c((rep(0,numCeros)),rnbinom(numcol-numCeros, mu=2,size=2)))}
+ if (distribucion==5)
+ {base[i,]<-sample(c((rep(0,numCeros)),rnbinom(numcol-numCeros, mu=20,size=1.5)))}
+ if (distribucion==6)
+ {base[i,]<-sample(c((rep(0,numCeros)),rnbinom(numcol-numCeros, mu=20,size=2)))}
+ if (distribucion==7)
+ {base[i,]<-sample(c(rPT(n = 1000, a = 0.5, mu = 2, D = 2)))}
+ if (distribucion==8)
+ {base[i,]<-sample(c(rPT(n = 1000, a = 0.5, mu = 2, D = 6)))}
+ if (distribucion==9)
+ {base[i,]<-sample(c(rPT(n = 1000, a = 0.5, mu = 2, D = 20)))}
+ if (distribucion==10)
+ {base[i,]<-sample(c(rPT(n = 1000, a = 0.5, mu = 5, D = 2)))}
```

```
+ if (distribucion==11)
+ {base[i,]<-sample(c(rPT(n = 1000, a = 0.5, mu = 5, D = 6)))}
+ if (distribucion==12)
+ {base[i,]<-sample(c(rPT(n = 1000, a = 0.5, mu = 5, D = 20)))}
+ if (distribucion==13)
+ {base[i,]<-sample(c(rPT(n = 1000, a = 0.5, mu = 10, D = 2)))}
+ if (distribucion==14)
+ {base[i,]<-sample(c(rPT(n = 1000, a = 0.5, mu = 10, D = 6)))}
+ if (distribucion==15)
+ {base[i,]<-sample(c(rPT(n = 1000, a = 0.5, mu = 10, D = 20)))}
+ }
+ base
+ }
```

simularAICs_P_valor(), es la función utilizada para extraer los AIC y p-valores de los modelos que nos da como resultado un objeto en el que se guarda todos los datos de la simulación

```
> simularAICs_P_valor<-function(base,factormodelo)
+ {
+ n<-length(base[1,])
+ {
+ Poip_value<-"NULL"
+ AICPoi<-"NULL"
+ SumPoi<-"NULL"
+ NBp_value<-"NULL"
+ AICNB<-"NULL"
+ AICPT<-"NULL"
+ PTp_value<-"NULL"
+ ZIPp_value<-"NULL"
+ ZIPp_value2<-"NULL"
+ AICZIP<-"NULL"
+ LRTZIP<-"NULL"
+ ZINBp_value<-"NULL"
+ ZINBp_value2<-"NULL"
+ AICZINB<-"NULL"
+ LRTZINB<-"NULL"
+ HPp_value<-"NULL"
+ HPp_value2<-"NULL"
+ AICHP<-"NULL"
+ LRTHP<-"NULL"
+ HNBp_value<-"NULL"
+ HNBp_value2<-"NULL"
+ AICHNB<-"NULL"
+ LRTHNB<-"NULL"
+ matrixaic<-"NULL"
```

```
+ matrixp_va<-"NULL"
+ modelos<-"NULL"
+ }
+ for(i in 1:length(base[,1]))
+ {
+ numzeros<-length(subset(base[i,],base[i,]==0))
+ formula<-base~factormodelo
+ datos<-data.frame(base=base[i,],factormodelo=factormodelo)
+ l<-length(formula)-2
+ #Modelos
+ modPoi <- glm(formula,family=poisson,data=datos)
+ modNB<-glm.nb(formula, data =datos)
+ modPT<-glmPT(formula, data = datos)
+ if(numzeros!=0)
+ {
+ modZIP<-zeroinfl(formula,data=datos, dist = "poisson")
+ modZIPr<-zeroinfl(base~1,data=datos, dist = "poisson")
+ modZINB<-zeroinfl(formula,data=datos, dist = "negbin")
+ modZINBr<-zeroinfl(base~1,data=datos, dist = "negbin")
+ modHP<-hurdle(formula,data=datos, dist = "poisson")
+ modHPr<-hurdle(base~1,data=datos, dist = "poisson")
+ modHNB<-hurdle(formula,data=datos, dist = "negbin")
+ modHNBr<-hurdle(base~1,data=datos, dist = "negbin")
+ }
+ #Parametros recogidos
+ #Poisson
+ SumPoi<-summary(modPoi)
+ Poip_value[i]<-round(SumPoi$coefficients[2,4],5)
+ AICPoi[i]<- AIC(modPoi)
+ #Binomial Negativa
+ SumNB<-summary(modNB)
+ NBp_value[i]<- round(SumNB$coefficients[2,4],5)
+ AICNB[i]<- AIC(modNB)
+ #Poisson Tweedie
+ SumPT<-summary(modPT)
+ PTp_value[i]<-SumPT$model.coef[2,4]
+ AICPT[i]<-AIC(modPT)
+ if(numzeros!=0)
+ {
+ #Zeroinflated poisson
+ SumZIP<-summary(modZIP)
+ ZIPp_value[i]<- SumZIP$coefficients$count[2,4]
+ ZIPp_value2[i]<- SumZIP$coefficients$zero[2,4]
+ AICZIP[i]<-AIC(modZIP)
```

```
+ a1<-lrtest(modZIPr,modZIP)
+ LRTZIP[i]<-round(a1[2,5],3)
+
+ #Zeroinflated Binomial Negativa
+ SumZINB<-summary(modZINB)
+ ZINBp_value[i]<- SumZINB$coefficients$count[2,4]
+ ZINBp_value2[i]<- SumZINB$coefficients$zero[2,4]
+ AICZINB[i]<-AIC(modZINB)
+ a2<-lrtest(modZINBr,modZINB)
+ LRTZINB[i]<-round(a2[2,5],3)
+
+ #Hurdle Poisson
+ SumHP<-summary(modHP)
+ HPp_value[i]<- SumHP$coefficients$count[2,4]
+ HPp_value2[i]<- SumHP$coefficients$zero[2,4]
+ AICHP[i]<-AIC(modHP)
+ a3<-lrtest(modHPr,modHP)
+ LRTHP[i]<-round(a3[2,5],3)
+
+ #Hurdle Binomial Negativa
+ SumHNB<-summary(modHNB)
+ HNBp_value[i]<- SumHNB$coefficients$count[2,4]
+ HNBp_value2[i]<- SumHNB$coefficients$zero[2,4]
+ AICHNB[i]<-AIC(modHNB)
+ a4<-lrtest(modHNBr,modHNB)
+ LRTHNB[i]<-round(a4[2,5],3)
+ modelos<-list(modelos,SumPoi,SumNB,SumPT)
+
+ print(i)#contador para saber en que estado esta la simulación
+ #ya que una simulacion puede tardar varias horas en completarse
+ }
+
+ }
+ #ordenamos los resultados y anadimos NAs
+ {
+
+ AICPoi<-sort(as.numeric(AICPoi))
+ length(AICPoi)<-length(base[,1])
+ AICNB<-sort(as.numeric(AICNB))
+ length(AICNB)<-length(base[,1])
+ AICPT<-sort(as.numeric(AICPT))
+ length(AICPT)<-length(base[,1])
+ AICZIP<-sort(as.numeric(AICZIP))
+ length(AICZIP)<-length(base[,1])
```

```
+ AICZINB<-sort(as.numeric(AICZINB))
+ length(AICZINB)<-length(base[,1])
+ AICHP<-sort(as.numeric(AICHP))
+ length(AICHP)<-length(base[,1])
+ AICHNB<-sort(as.numeric(AICHNB))
+ length(AICHNB)<-length(base[,1])
+ Poip_value<-as.numeric(Poip_value)
+ NBp_value<-as.numeric(NBp_value)
+ PTp_value<-as.numeric(PTp_value)
+ ZIPp_value<-as.numeric(ZIPp_value)
+ ZIPp_value2<-as.numeric(ZIPp_value2)
+ LRTZIP<-as.numeric(LRTZIP)
+ ZINBp_value<-as.numeric(ZINBp_value)
+ ZINBp_value2<-as.numeric(ZINBp_value2)
+ LRTZINB<-as.numeric(LRTZINB)
+ HPp_value<-as.numeric(HPp_value)
+ HPp_value2<-as.numeric(HPp_value2)
+ LRTHP<-as.numeric(LRTHP)
+ HNBp_value<-as.numeric(HNBp_value)
+ HNBp_value2<-as.numeric(HNBp_value2)
+ LRTHNB<-as.numeric(LRTHNB)
+
+ }
+
+
+
+
+ matrixaic<-data.frame(Poi=AICPoi, NB=AICNB, PT=AICPT, ZIP=AICZIP, ZINB=AICZINB,HP=AICHP,
+                       HNB=AICHNB  )
+ matrixp_va<-data.frame(nsimu=length(base[,1]),
+     Poi=round(mean(Poip_value<0.05,na.rm=T),3),NB=round(mean(NBp_value<0.05,na.rm=T),2),
+     PT=round(mean(PTp_value<0.05,na.rm=T),3),
+     ZIPz=round(mean(ZIPp_value2<0.05,na.rm=T),3),ZIP=round( mean(ZIPp_value<0.05,na.rm=T),2),
+     RTZIP=round(mean(LRTZIP<0.05,na.rm=T),3),
+         ZINBz=round(mean(ZINBp_value2<0.05,na.rm=T),3), ZINB=round(mean(ZINBp_value<0.05,na.rm=T),3),
+     LRTZINB=round(mean(LRTZINB<0.05,na.rm=T),3),
+     HPz=round(mean(HPp_value2<0.05,na.rm=T),3),HP=round(mean(HPp_value<0.05,na.rm=T),3),
+     LRTHP=round(mean(LRTHP<0.05,na.rm=T),3),
+     HNBz=round(mean(HNBp_value2<0.05,na.rm=T),3),HNB=round(mean(HNBp_value<0.05,na.rm=T),3),
+     LRTHNB=round(mean(LRTHNB<0.05,na.rm=T),3)
+ )
+ res<-list(matrixaic=matrixaic,      matrixp_va=matrixp_va,
+           Poi=Poip_value,     NB=NBp_value,      PT=PTp_value,
+           ZIPz=ZIPp_value2,   ZIP=ZIPp_value,    LRTZIP=LRTZIP,
```

```
+                   ZINBz=ZINBp_value2, ZINB=ZINBp_value,  LRTZINB=LRTZINB,
+                   HPz=HPp_value2 ,    HP=HPp_value,      LRTHP=LRTHP,
+           HNBz=        HNBp_value2, HNB=HNBp_value,     LRTHNB=LRTHNB,
+                   base=base,modelos=modelos)
+
+ res
+ }
>
```

La función `simulapower()` nos proporciona los datos necesarios para crear las curvas de potencia.

Podemos escoger la n de cada grupo, el numero de simulaciones que hace con cada parametro, la $\mu$ inicial y la $\mu$ final, y el parámetro de dispersión y parámetro de forma $a$ de la distribución PT.

```
> simulapower<-function(n,nsimu,mu1,mu2,fi,a,paso=NULL)
+ {
+
+ set.seed(123)
+
+ NB<-"NULL"
+ PT<-"NULL"
+ ZINB<-"NULL"
+ ZINBz<-"NULL"
+ HNB<-"NULL"
+ HNBz<-"NULL"
+ diftotal<-mu2-mu1
+ NBp_value<-matrix(rep(0,nsimu*diftotal),nsimu,diftotal)
+ PTp_value<-matrix(rep(0,nsimu*diftotal),nsimu,diftotal)
+ ZINBp_value<-matrix(rep(0,nsimu*diftotal),nsimu,diftotal)
+ ZINBp_value2<-matrix(rep(0,nsimu*diftotal),nsimu,diftotal)
+ HNBp_value<-matrix(rep(0,nsimu*diftotal),nsimu,diftotal)
+ HNBp_value2<-matrix(rep(0,nsimu*diftotal),nsimu,diftotal)
+ diftotal<-mu2-mu1
+ for(j in 1:diftotal)
+ {
+ base<-matrix(c(rep(0,2*n*nsimu),0,0),(n*2),nsimu)
+ factormodelo<-c(rep("A",n),rep("B",n))
+ for(i in 1:nsimu)
+ {
+ base[,i]<-c(rPT(n =n, a =a, mu =mu1, D = fi,max=1000),
+             rPT(n =n, a =a, mu =mu1+j, D = fi,max=1000))
+ }
+ #Hacer modelos
+ for(i in 1:length(base[1,]))
```

```
+ {
+ formula<-base~factormodelo
+ datos<-data.frame(base=base[,i],factormodelo=factormodelo)
+
+ #Modelos
+ modPT<-glmPT(formula, data = datos)
+ modNB<-glm.nb(formula, data =datos)
+ numzeros<-length(subset(base[,i],base[,i]==0))
+ if(numzeros!=0)
+ {
+ modZINB<-zeroinfl(formula,data=datos, dist = "negbin")
+ modHNB<-hurdle(formula,data=datos, dist = "negbin")
+ }
+ #Binomial Negativa
+ SumNB<-summary(modNB)
+ NBp_value[i,j]<- SumNB$coefficients[2,4]
+ #Poisson Tweddie
+ SumPT<-summary(modPT)
+ PTp_value[i,j]<-as.numeric(SumPT$model.coef[2,4])
+ if(numzeros!=0)
+ {
+ #Zeroinflated Binomial Negativa
+ SumZINB<-summary(modZINB)
+ ZINBp_value[i,j]<- SumZINB$coefficients$count[2,4]
+ ZINBp_value2[i,j]<- SumZINB$coefficients$zero[2,4]
+
+ #Hurdle Binomial Negativa
+ SumHNB<-summary(modHNB)
+ HNBp_value[i,j]<- SumHNB$coefficients$count[2,4]
+ HNBp_value2[i,j]<- SumHNB$coefficients$zero[2,4]
+ }
+ print(i)
+ print(j)
+ print(diftotal)
+ }
+
+ }
+ NB<-colMeans(NBp_value<0.05,na.rm=T)
+ PT<-colMeans(PTp_value<0.05,na.rm=T)
+ ZINB<-colMeans(ZINBp_value<0.05,na.rm=T)
+ ZINBz<-colMeans(ZINBp_value2<0.05)
+ HNB<-colMeans(HNBp_value<0.05)
+ HNBz<-colMeans(HNBp_value2<0.05)
+ res<-list(NBp_value=NBp_value,PTp_value=PTp_value,ZINBp_value=ZINBp_value,
```

```
+            ZINBp_value2=ZINBp_value2,HNBp_value,HNBp_value2=HNBp_value2,
+           parametros=data.frame(n=n,nsimu=nsimu,mu1=mu1,mu2=mu2,fi=fi,a=a),
+           porcentajes=data.frame(PT=PT,NB=NB,ZINB=ZINB,ZINBz=ZINBz,
+ HNB=HNB,HNBz=HNBz)
+            )
+ res
+
+ }
```

Datos Pickrell.

Cargamos los datos y visualizamos los datos correspondientes al gen número 11 que es sobre el que realizamos el estudio individualizado.

```
> load("pickrell.rda")
> countsNigerian <- exprs(pickrell.eset)
> dim(countsNigerian)
> gender <- pickrell.eset$gender
> #Limpieza de todos ceros
> datos<-cbind(pickrell,rowMeans(pickrell))
> dim(datos)
> datos<-subset(datos,datos[,70]!=0)
> datos<-datos[,-70]
> dim(datos)
> head (datos)
> datos[11,]
```

Codigo del análisis para el gen 11

```
> xf <- unlist(datos[11, gender=="female"])
> xm <- unlist(datos[11, gender=="male"])
> xf
> xm
> sort(xf)
> sort(xm)
> #gráficas de comparación de las distribución empírica y estimada
> compareCountDist(xf, main="Female samples")

> compareCountDist(xm, main="Male samples")
```

Guardamos las gráficas en pdf.

Realizamos los modelos del gen 11 y miramos los resultados

```
> dat<-datos[11,]
> modPoisson <- glm(dat~gender,family=poisson)
> modNB<-glm.nb(dat~gender)
> modPT<-glmPT(dat~gender)
```

```
> modHNB<-hurdle(dat~gender, dist = "negbin")
> summary(modPoisson)
> modPoisson$coefficients
> summary(modNB)
> summary(modPT)
> summary(modHNB)
```

Figura 5.5, mediante esta sintaxis observamos que siempre el AIC del modelo PT es menor

```
> load("res_pt_0.5_2_20.rda")
> v<-res_pt_0.5_2_20
> h=NULL
> b=NULL
> for(i in 1:500)
+ {
+ h[i]<-min(v$matrixaic[i,])
+
+ if( h[i]== v$matrixaic[i,3]){b[i]="PT"}
+
+ else  b[i]="otra"
+
+
+ }
> b
```