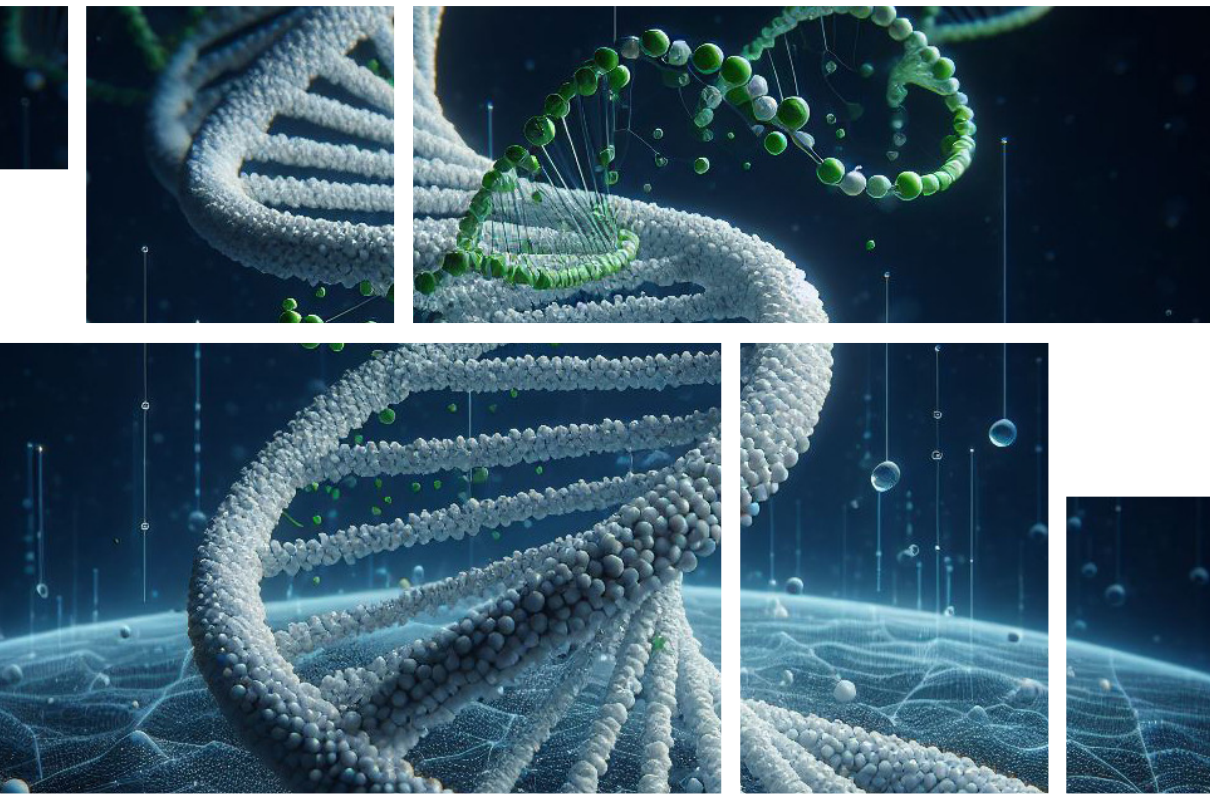
DOCUMENTOS
244

e-ISSN 2965-8047
Novembro/2023

OBJETIVOS DE DESENVOLVIMENTO SUSTENTÁVEL

4 EDUCAÇÃO DE QUALIDADE

Manual para identificação de RNAs longos não codificantes (lncRNAs) por meio de ferramentas bioinformáticas na análise de transcriptomas



inct institutos nacionais de ciência e tecnologia

UFV Universidade Federal de Viçosa

Embrapa

e-ISSN 2965-8047
Novembro/2023

Empresa Brasileira de Pesquisa Agropecuária
Embrapa Suínos e Aves
Ministério da Agricultura, Pecuária e Abastecimento

DOCUMENTOS 244

Manual para identificação de RNAs longos não codificantes (lncRNAs) por meio de ferramentas bioinformáticas na análise de transcriptomas

Francelly Geralda Campos
Adriana Mércia Guaratini Ibelli,
Haniel Cedraz de Oliveira
Maurício Egídio Cantão
Jane de Oliveira Peixoto
Susana Amaral Teixeira
Karine Assis Costa
Mônica Corrêa Ledur
Simone Eliza Facioni Guimarães

Autores

Embrapa Suínos e Aves
Concórdia, SC
2023

Exemplares desta publicação podem ser adquiridos na:

Embrapa Suínos e Aves
Rodovia BR 153 - KM 110
Caixa Postal 321
89.715-899, Concórdia, SC
Fone: (49) 3441 0400
Fax: (49) 3441 0497
www.embrapa.br
www.embrapa.br/fale-conosco/sac

Supervisão editorial
Tânia Maria Biavatti Celant

Revisão técnica
Roberto Hiroshi Higa
Simone Cristina Méo Niciura

Revisão de texto
Jean Carlos Porto Vilas Boas Souza

Normalização bibliográfica
Claudia Antunes Arrieche

Projeto gráfico da coleção
Carlos Eduardo Felice Barbeiro

Editoração eletrônica
Vivian Fracasso

Foto da capa
Maurício Cantão (gerada no BING)

1ª edição
Versão eletrônica (2023)

Dados Internacionais de Catalogação na Publicação (CIP)
Embrapa Suínos e Aves

Manual para identificação de RNAs longos não codificantes (lncRNAs) por meio de ferramentas bioinformáticas na análise de transcriptomas / Francelly Geralda Campos. – Concórdia : Embrapa Suínos e Aves, 2023.

41 p.; 21 cm. (Documentos / Embrapa Suínos e Aves, e-ISSN 2965-8047; 244).

1. Suíno. 2. Genoma. 3. Sequenciamento. 4. Expressão Gênica. 5. Análise de lncRNAs. 6. Tecnologia. 7. Bioinformática. 8. Manual. I. Título. II. Série. III. Campo, Francelly Geralda. IV. Ibelli, Adriana Mércia Guaratini. V. Oliveira, Haniel Cedraz de. VI. Cantão, Maurício Egídio. VII. Peixoto, Jane de Oliveira. VII. Amaral, Susana Teixeira. VIII. Costa, Karine Assis. IX. Ledur, Mônica Corrêa. X. Guimarães, Simone Eliza Facioni.

CDD. 572.86

Claudia Antunes Arrieche - CRB 14/880

Autor

Francelly Geralda Campos

Zootecnista, mestre em Zootecnia, doutoranda do Programa de Pós-Graduação em Zootecnia, Universidade Federal de Viçosa, Viçosa, MG

Adriana Mércia Guaratini Ibelli

Bióloga, doutora em Genética Evolutiva e Biologia Molecular, analista da Embrapa Suínos e Aves, Concórdia, SC

Haniel Cedraz de Oliveira

Médico Veterinário, doutor em Zootecnia, Programa de Pós-Graduação em Zootecnia, Universidade Federal de Viçosa, Viçosa, MG

Maurício Egídio Cantão

Técnologo em Processamento de Dados, doutor em Bioinformática, pesquisador da Embrapa Suínos e Aves, Concórdia, SC

Jane de Oliveira Peixoto

Zootecnista, doutora em Zootecnia, pesquisadora da Embrapa Suínos e Aves, Concórdia, SC

Susana Amaral Teixeira

Zootecnista, doutora em Zootecnia, professora adjunta na Universidade Federal do Mato Grosso do Sul, Campo Grande, MS

Karine Assis Costa

Bióloga, doutora em Genética e Melhoramento, Programa de Pós-Graduação em Zootecnia, professora na Universidade Estadual Paulista "Júlio de Mesquita Filho", Ilha Solteira, SP

Mônica Corrêa Ledur

Zootecnista, doutora em Genética e Melhoramento Animal, pesquisadora da Embrapa Suínos e Aves, Concórdia, SC

Simone Eliza Facioni Guimarães

Médica Veterinária, doutora em Ciência Animal, professora do Departamento de Zootecnia da Universidade Federal de Viçosa, Viçosa, MG

Apresentação

O projeto genoma humano, que envolveu laboratórios e pesquisadores do mundo inteiro, foi um marco inicial importante para o progresso de metodologias de sequenciamento, criação de bancos de dados e o desenvolvimento de ferramentas de bioinformática para a análise de grande volume de informações que começou a ser gerado principalmente após a década de 2010 para várias espécies de organismos.

Atualmente, as ferramentas das Ciências Ômicas são utilizadas nos mais variados setores, incluindo os relacionados aos animais de produção. Dentre suas aplicações na produção animal estão a identificação de biomarcadores associados às doenças, a melhor compreensão da biologia básica da fisiologia dos organismos e da arquitetura genética de características economicamente importantes e também a seleção mais acurada de animais com maior potencial genético.

Neste contexto, as ferramentas de transcriptômica são utilizadas para se ter o conhecimento do conjunto completo de transcritos (RNAs mensageiros, ribossômicos, transportadores e não codificantes) de um dado organismo, órgão, tecido ou linhagem celular em circunstâncias específicas (em decorrência de diferentes condições ambientais ou experimentais, entre outras situações). Assim, a utilização de ferramentas de bioinformática é indispensável para realizar análises de sequenciamento de RNA e de outras ômicas.

A identificação de RNAs longos não codificantes (lncRNAs) e de suas funções é importante para compreender como a expressão gênica é regulada e sua consequência na determinação de fenótipos. Deste modo, dada a escassez de informações em português sobre este tipo de análise, este manual

traz o passo-a-passo para identificar lncRNAs a partir de dados de RNA-Seq, utilizando como exemplo os suínos. A disponibilização dessas informações de uma maneira simplificada permite uma melhor compreensão dos tipos de análise necessários, possibilitando ao leitor realizar as análises seguindo o protocolo descrito

Everton Luis Krabbe
Chefe geral da Embrapa Suínos e Aves

Sumário

Introdução

O desenvolvimento de tecnologias de sequenciamento de nova geração, como RNA-seq, possibilita melhorar a compreensão da estrutura dos genes e dos padrões de expressão gênica. Essas tecnologias e os métodos computacionais tornam possível o estudo mais profundo do transcriptoma, aplicados para a identificação e caracterização dos RNAs codificantes e não codificantes. O sequenciamento permite caracterizar as isoformas de genes conhecidos e melhorar a anotação e identificação de novos genes e transcritos não codificantes (Esteve-Codina et al., 2011). Para maximizar a obtenção de resultados acurados a partir do sequenciamento de transcriptoma é necessário adequar o protocolo para a preparação das bibliotecas, as quais podem ser restritas ao RNA mensageiro (mRNA) ou podem amplificar também outros tipos de transcritos, como quando se utiliza os kits de depleção de RNA ribossomal (rRNA). A depleção de rRNA permite a detecção eficiente de transcritos codificantes e não codificantes (Herbert et al., 2018; Vecera et al., 2019). Os RNAs não codificantes são divididos em duas classes: RNAs pequenos não codificantes (menos de 200 nt), que incluem microRNAs, RNAs de interação com piwi (piRNA) e RNAs nucleolares pequenos; e os RNAs longos não codificantes (lncRNAs) (mais de 200 nt) (Taft et al., 2010; Dozmorov et al., 2013; Zhang, et al., 2019).

Os lncRNAs compartilham muitas características análogas aos mRNA, como estruturas multiexônicas, sítio 5' Cap e poliadenilação, porém não possuem potencial de codificação. Nos últimos anos, com o avanço da genômica e das ferramentas de bioinformática, tem sido possível identificar milhares de lncRNAs em humanos e animais (Mattick; Rinn, 2014, Li; Liu, 2019, Li; Zhang; Liu, 2020). Embora as funções da maioria dos lncRNAs permaneçam desconhecidas, diversos estudos relataram que os lncRNAs geralmente exibem padrões de expressão específicos do tecido ou do estágio de desenvolvimento e estão envolvidos em uma ampla gama de funções, incluindo modificação da cromatina (Rinn et al., 2007), *imprinting* (Sleutels; Zwart; Barlow, 2002), transcrição (WANG et al., 2008), *splicing* (Yan et al., 2005), processamento pós-transcricional (He et al., 2008; Zang et al., 2010) e tradução (Wang et al., 2005).

Para prever as funções dos lncRNAs é preciso analisar sua ação *cis* ou *trans* em relação aos genes codificadores de proteínas com funções conhecidas (Yan et al. 2017). A ação cis regula a transcrição de genes próximos. Em contraste, os fatores transregulatórios regulam (ou modificam) a expressão de genes distantes combinando-se com suas sequências alvo (Wittkopp et al. 2004). Os lncRNAs não têm o mesmo padrão de conservação interespécies que os genes codificadores de proteínas. A falta de estudos funcionais e a baixa conservação de sequências tornam a interpretação funcional desses transcritos um desafio para as pesquisas (Johnsson et al. 2014).

O aumento dos estudos com RNAs não codificantes (Figura 1), especialmente com as iniciativas do projeto Encode (https://www.encodeproject.org/), possibilitou o conhecimento e entendimentos dos mesmos. Embora existam vários lncRNAs bem caracterizados, poucos protocolos do uso das ferramentas para a predição dos lncRNAs estão disponíveis, especialmente quando se procura material em português. Desta forma, há a necessidade do desenvolvimento de documentos que auxiliem na utilização destas ferramentas de predição, facilitando o entendimento tanto das análises quanto da interpretação de resultados.

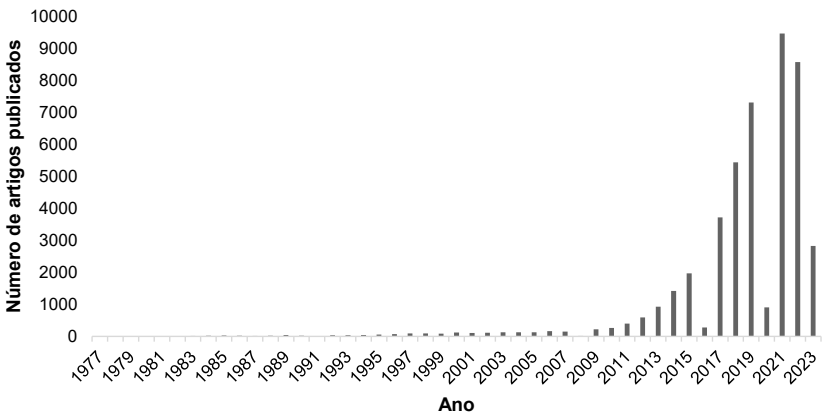


Figura 1. Número de artigos publicados utilizando análises de lncRNAs desde a década de 1970 até 2023, de acordo com o PubMed. Termo de pesquisa: lncRNA.

Fonte: NCBI, 2023 (acesso em 15 de maio de 2023).

Desta forma, o objetivo deste documento é fornecer informações para as análises de lncRNAs, em português, de forma detalhada e prática, tendo como público alvo estudantes de graduação, pós-graduação e cientistas que pretendam utilizar estas abordagens de análises de expressão gênica em suínos ou outras espécies de animais. Para isso, são apresentadas as etapas para realização dessa análise e as ferramentas empregadas em cada uma dessas etapas.

O documento está organizado da seguinte forma: (i) na seção 2 são apresentadas as etapas e principais ferramentas utilizadas em cada uma delas, para que o leitor tenha uma visão geral da análise apresentada; e (i) na seção 3 essas ferramentas são aplicadas, ou via scripts ou diretamente via comando shell, em um passo-a-passo que ilustra como realizar esse tipo de análise, utilizando um conjunto de dados de RNA sequenciado (RNA-Seq) de suíno. É a sequência de comandos apresentadas nessa seção, com possíveis adaptações pontuais, que o leitor deve executar na análise dos seus dados.

Ferramentas para identificação de lncRNAs

Para realizar as análises transcriptômicas descritas neste documento, são necessários dados de sequenciamento de RNAs longos que em geral são obtidos através de bibliotecas de RNA preparadas com protocolo de depleção de RNA ribossomal (rRNA). Neste manual são utilizados arquivos brutos de sequências no formato FASTQ e as análises são realizadas em ambiente Linux, utilizando a linguagem de programação R e Shell.

As metodologias analíticas para identificar lncRNAs podem ser classificadas em duas categorias: análises com ou sem o uso do genoma de referência. O foco deste documento está em abordar a metodologia de análise utilizando um genoma de referência disponível, e os passos utilizados são: controle de qualidade (CQ), mapeamento, montagem, filtragem, potencial de codificação e expressão diferencial. A seguir, será relatado um breve resumo dos programas utilizados para realização de cada etapa de análise dos lncRNAs em um conjunto de dados de sequenciamento de RNA provenientes de amostras de suínos.

Controle de qualidade

O controle de qualidade tem por objetivo filtrar as sequências de baixa qualidade, removendo assim erros de leitura. Existem ferramentas disponíveis na literatura para esta etapa, como o Trimmomatic (Bolger et al., 2014) ou Cutadapt (Martin, 2011). Neste tutorial, utilizou-se no controle de qualidade das sequências a ferramenta Trimmomatic para o controle de qualidade das sequências para remoção de adaptadores, e de bases de baixa qualidade do início e do fim com baixa qualidade ou Ns, para os quais não foi possível atribuir uma base. Para essa análise utilizou-se os seguintes parâmetros do programa: remoção das bases quando a qualidade Phred média por base é menor do que 15 e descarte de leituras (reads) com comprimento menor do que 20 bases. Esta ferramenta de pré-processamento é flexível e eficiente, lidando com dados pareados (Bolger et al., 2014). Para iniciar a utilização do Trimmomatic é necessário utilizar arquivos do tipo FASTQ (Figura 2). Esse arquivo é projetado para lidar com a saída de métricas de qualidade básica de sequenciamento. As pontuações de sequência e qualidade são representadas como caracteres ASCII simples. O formato usa quatro linhas para cada sequência. A primeira linha tem o título com o caractere '@' e é seguida por um identificador de sequência e uma descrição opcional. A segunda linha é a sequência. A terceira marca o fim da sequência com o caractere '+'. A quarta linha contém os valores de qualidade do sequenciamento para cada base, em que o valor do caractere ASCII, ao ser subtraído por 33, corresponde ao valor de qualidade Phred.(Figura 3).

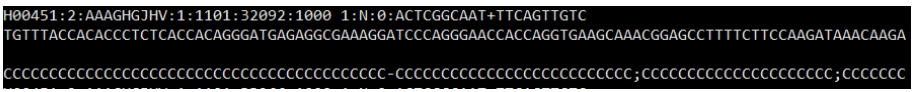
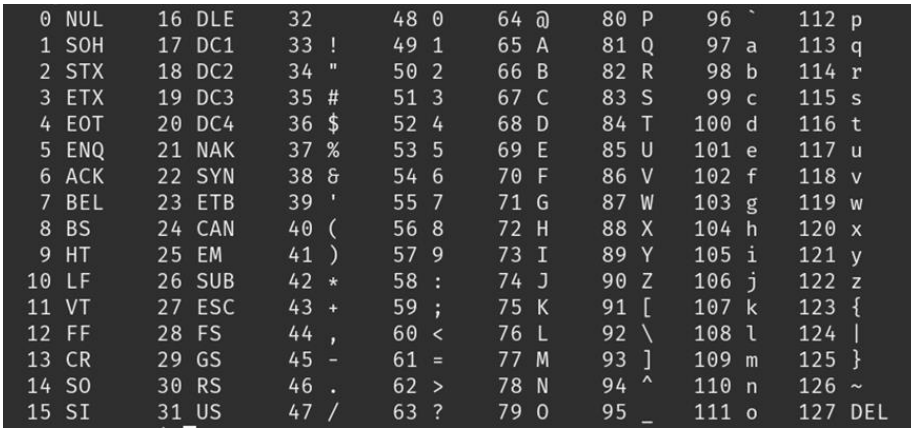


Figura 2. Representação de um arquivo FASTQ.

As quatro linhas representam uma leitura. Uma leitura (*read*) consiste em uma saída de uma única sequência de nucleotídeos de uma máquina de sequenciamento. Uma leitura pode consistir em vários segmentos. Para dados de sequenciamento, as leituras são indexadas pela ordem em que são sequenciadas e podem ser longas, curtas, emparelhadas ou únicas. A limpeza e filtragem das leituras é uma etapa importante para o pré-processamento dos dados brutos de sequenciamento.



0	NUL	16	DLE	32		48	0	64	@	80	P	96	`	112	p
1	SOH	17	DC1	33	!	49	1	65	A	81	Q	97	a	113	q
2	STX	18	DC2	34	"	50	2	66	B	82	R	98	b	114	r
3	ETX	19	DC3	35	#	51	3	67	C	83	S	99	c	115	s
4	EOT	20	DC4	36	$	52	4	68	D	84	T	100	d	116	t
5	ENQ	21	NAK	37	%	53	5	69	E	85	U	101	e	117	u
6	ACK	22	SYN	38	&	54	6	70	F	86	V	102	f	118	v
7	BEL	23	ETB	39	'	55	7	71	G	87	W	103	g	119	w
8	BS	24	CAN	40	(	56	8	72	H	88	X	104	h	120	x
9	HT	25	EM	41	)	57	9	73	I	89	Y	105	i	121	y
10	LF	26	SUB	42	*	58	:	74	J	90	Z	106	j	122	z
11	VT	27	ESC	43	+	59	;	75	K	91	[	107	k	123	{
12	FF	28	FS	44	,	60	<	76	L	92	\	108	l	124	|
13	CR	29	GS	45	-	61	=	77	M	93	]	109	m	125	}
14	SO	30	RS	46	.	62	>	78	N	94	^	110	n	126	~
15	SI	31	US	47	/	63	?	79	O	95	_	111	o	127	DEL

Figura 3. Caracteres da tabela ASCII.

Mapeamento

Após o controle de qualidade, as sequências filtradas (limpas) de cada amostra são mapeadas contra o genoma de referência (.fa ou .fna), podendo ou não utilizar as informações de anotação (.gff3 ou .gtf). No caso dos arquivos utilizados neste documento, utilizamos o genoma do suíno (*Sus scrofa11.1*) na anotação Ensembl com o software HISAT2 (Kim; Langmead; Salzberg, 2015). Os genomas de referência de diferentes espécies podem ser encontrados nas bases de dados *Genome* do NCBI (https://www.ncbi.nlm.nih.gov/data-hub/genome/), Ensembl (https://www.ensembl.org/index.html) ou do UCSC (https://genome.ucsc.edu/). É importante reiterar que os arquivos de genoma (.fa) e de anotação (.gff3 ou .gtf) devem sempre ser baixados de uma mesma base de dados, pois pode haver diferenças entre as plataformas.

O HISAT2 é um programa de alinhamento rápido e sensível para mapeamento de leituras de RNA-seq, sendo um programa amplamente utilizado. É uma versão atualizada do famoso *TopHat* e requer menos recursos computacionais do que outros programas (Lui et al., 2022). A entrada do HISAT2 é composta por um conjunto de sequências em arquivos FASTQ e a saída por outro conjunto de arquivos no formato SAM. Este é um formato padronizado para apresentar dados de sequência alinhados. Porém, este é um arquivo de texto simples que tende a ser muito grande e, por isso, muitas vezes é convertido em arquivos BAM, a versão binária (compactada) dos arquivos SAM. Esses

arquivos são muito menores e podem ser indexados, o que permite tanto acessá-los quanto processá-los com maior eficiência (Ramirez-Gonzalez et al., 2012).

Em seguida ao mapeamento, normalmente, nas análises de lncRNA, são feitas as montagens de transcritos a fim de verificar a presença de novos transcritos (Pertea et al., 2015).

Montagem dos transcritos usando StringTie

O StringTie é um montador rápido e altamente eficiente para mapeamentos de RNA-Seq em potenciais transcritos. Ele usa um algoritmo de fluxo de rede, bem como uma etapa de montagem *de novo* opcional para montar e quantificar transcrições completas que representam múltiplas variantes de *splicing* para cada *locus* de gene. A montagem de genes e suas isoformas usando o StringTie é um processo que utiliza o mapeamento das reads contra o genoma de referência e realiza a reconstrução dos transcritos por meio das regiões mapeadas. Após o processo de montagem dos transcritos, o StringTie estima os níveis de expressão de todos os genes e isoformas (Pertea et al., 2015). O StringTie recebe como entrada arquivos SAM, BAM ou CRAM com alinhamentos de leitura de RNA-Seq ordenados por sua localização genômica e a saída é um arquivo gtf para cada amostra contendo os transcritos montados. Adicionalmente aos transcritos montados, é possível compilar os resultados da montagem com a anotação disponível do genoma. Para isso, utiliza-se um programa do pacote StringTie, o *gffcompare*, que compara os transcritos montados com a anotação do genoma de referência e cria uma versão única e completa de todos os transcritos do genoma. Para mais informações sobre o StringTie, consulte sua documentação em http://ccb.jhu.edu/software/stringtie/index.shtml?t=manual.

Filtragem de transcritos

Após a montagem com o StringTie, para maior precisão de anotação dos lncRNAs, é realizada uma filtragem rigorosa. Primeiramente, antes de fazer a mesclagem das amostras, são removidos transcritos com valor de FPKM (Fragmentos por Kilobase por Milhão) inferior a 0.5. Após a junção das amos-

tras, são removidos transcritos que apresentem classe de codificação conhecidas, transcritos com éxon único e com tamanho ≤ 200 nucleotídeos.

Potencial de codificação

Identificar o potencial de codificação de um novo transcrito é importante para a anotação precisa e análise correta dos lncRNAs (Kong et al., 2007). Aqui estão detalhadas algumas ferramentas usadas para prever o potencial de codificação de transcritos. Para mais informações, consulte a literatura citada.

CPC2 (*Coding Potential Calculator*)

Uma das primeiras ferramentas disponíveis para identificar o potencial de codificação de um novo transcrito é o CPC (*Coding Potential Calculator*). Os recursos utilizados pelo modelo preditivo do CPC incluem a cobertura da ORF (*Open Reading Frame*) e a semelhança da sequência com genes codificadores de proteínas conhecidos (Kong et al., 2007). Esses recursos foram incorporados em um classificador de aprendizado de máquina SVM (*Support Vector Machine*), que em 2017 foi atualizado e incorporado ao *Coding Potential Calculator* 2 (CPC2), que avalia o potencial de codificação do transcrito com base nas características das sequências biológicas, incluindo homologia com sequências de proteínas conhecidas (Kang et al., 2017). O CPC2 está disponível gratuitamente em http://cpc2.cbi.pku.edu.cn como um servidor da web e um pacote independente para download.

CNCI (*Coding-Non-Coding Index*)

O CNCI possui duas etapas principais: a construção do modelo de classificação e a pontuação da sequência. Este programa realiza a classificação das sequências em codificantes e não codificantes de proteínas, para dados de sequenciamento de alto rendimento, por meio do perfil de trinca de nucleotídeos independentes. Quando a pontuação é menor que zero, a transcrição é considerada RNA não codificante. Ele também prevê transcrições incompletas e antisense (Sun et al., 2013).

CPAT (*Coding Potential Assessment Tool*)

O CPAT determina a capacidade de codificação e não codificação de uma transcrição construindo um modelo de regressão logística com base no comprimento e cobertura da ORF e calculando as pontuações de Fickett e Hexamer. A pontuação de Fickett é obtida calculando as composições percentuais de A, C, G, T e seus valores de posição. O valor da posição reflete o grau em que cada base é favorecida em uma posição de códon em relação a outra. Esses oito valores são, então, convertidos em probabilidades (p) de codificação. Cada probabilidade é multiplicada por um peso (w) para a respectiva base, onde o valor de w reflete a chance de cada parâmetro sozinho predizer com sucesso a função codificante ou não codificadora para as sequências. A pontuação de Hexamer é calculada usando o logarítimo da razão de probabilidade para medir o uso de hexâmeros randômicos diferenciais entre as sequências codificantes e não codificantes (Wang et al., 2013).

PLEK (*Predictor of long non-coding RNAs and messenger RNAs based on an improved k-mer scheme*)

PLEK usa um pipeline computacional baseado em um k-mer e um algoritmo de SVM para distinguir lncRNAs de mRNAs. *K-mer* são substrings (uma sequência de caracteres dentro de uma sequência maior) de comprimento k em uma determinada string (podendo ser DNA, RNA, proteína ou outros) (Marçais; Kingsford, 2011). Ele é uma ferramenta livre de alinhamento com base nas frequências *k-mer* das sequências, considerando que cada janela de leitura é um nucleotídeo, e o uso de *k-mer* é calibrado de acordo com o tamanho em nucleotídeos de *k-mer* cadeias.

O PLEK é adequado para dados de transcriptoma com erros de sequenciamento *indel* e conjuntos de dados de transcriptoma em larga escala. Assim, ele é útil para distinguir sequências codificantes e não codificantes de proteínas a partir de dados de sequenciamento de alto rendimento, sendo uma ferramenta robusta e rápida para identificar lncRNAs. (Li; Zhang; Zhou, 2014).

Pfam *(database of protein families)*

O banco de dados Pfam é um sistema de classificação para a anotação de domínios de proteínas e é usado para identificar domínios de famílias de proteínas conhecidas. Ele estabelece um modelo estatístico (hidden Markov model - HMM) da sequência de aminoácidos de cada família através do alinhamento da sequência de proteínas. Um transcrito sem correspondência no Pfam é considerado um potencial lncRNA (Mistry et al., 2021).

Passo-a-passo para a identificação dos lncRNAs em suínos

Esta seção apresenta um procedimento passo-a-passo que ilustra como realizar a análise para identificação de lncRNAs, utilizando um conjunto de dados de RNA sequenciado (RNA-Seq) de um experimento com fetos de suínos (*Sus scrofa*). As amostras são provenientes de bibliotecas de RNA preparadas usando o kit *Illumina Stranded Total RNA Prep, Ligation with Ribo-Zero Plus* para depleção de RNA ribossomal e foram sequenciadas com a plataforma Illumina HiSeq2500. Todos os dados necessários para execução dessas análises estão disponíveis no link https://rb.gy/poz3b. Os comandos correspondentes a cada uma das etapas da análise devem ser executados a partir do prompt do shell do linux. Em todos os casos, será apresentado uma descrição dos parâmetros do comando e da saída (arquivo) resultante. Em relação às necessidades computacionais, sugere-se um mínimo de 24 processadores e ambiente de programação R.

O diretório de entrada contém um arquivo fastq correspondente a amostras coletadas de conceptos suínos, sendo uma amostra do sexo feminino (pig_01) e outra do sexo masculino (pig_02). Primeiramente, é necessário criar um diretório para armazenar todos os programas e amostras usados neste documento. É importante que o nome do arquivo/diretório não possua acentos nem espaços.

```bash
“‘bash“‘
$ mkdir Analise_lncRNAs
```

##Controle de qualidade e mapeamento

O pipeline BAQCOM é utilizado para realizar o controle de qualidade com o Trimmomatic e o mapeamento com HISAT2. O BAQCOM é um pipeline que automatiza a execução de cinco programas para análise de RNA-Seq. Para baixar e instalar o BAQCOM visite: https://github.com/hanielcedraz/BAQCOM.

#Gerar os arquivos de samples.txt

O BAQCOM disponibiliza um script para a criação do arquivo de amostras. Esse arquivo é composto por três colunas, separadas por tabulação, (SAMPLE_ID, Read_1 e Read_2) para arquivos *paired-end* ou duas colunas (SAMPLE_ID e Read_1) para arquivos *single-end*. Para a execução do comando é necessário criar uma pasta chamada 00-Fastq ou especificar no comando o nome da pasta.

```
‘‘‘bash‘‘‘ $ createSamples.sh
## Para mais informações acesse a ajuda usando a tag -h createSamples.sh -h
```

#Controle de qualidade – baqcomTrimmomatic.R

O baqcomTrimmomatic.R usa o software Trimmomatic para realizar o controle de qualidade dos arquivos e remove as leituras curtas, de baixa qualidade (QPhred < 20) e sequências de adaptadores, usando os seguintes parâmetros:

- **-q** => Número de amostras a serem processadas a cada vez.
- **-m** => Tamanho mínimo de sequência (<70 bp).
- **-p** => Número de processadores.

```
‘‘‘bash‘‘‘
$ baqcomTrimmmomatic.R –p 16 –q 2 –m 70
## Para mais informações acesse a ajuda usando a tag -h
$ baqcomTrimmmomatic.R -h
```

#Mapeamento - HISAT2 - baqcomHisat2.R

Os arquivos de saída do baqcomTrimmomatic serão usados como entrada para o baqcomHisat2.R usando o software HISAT2. Além disso, é necessário o uso dos arquivos fasta e gtf do genoma de referência de suínos. Esses arquivos podem ser encontrados em (http://ftp.ensembl.org/pub/release107/fasta/sus_scrofa/dna/Sus_scrofa.Sscrofa11.1.dna.toplevel.fa.gz e http://ftp.ensembl.org/pub/release-107/gtf/sus_scrofa/Sus_scrofa.Sscrofa11.1.107.gtf.gz), sendo que os mesmos devem ser descompactados.

```bash
$ gunzip Sus_scrofa.Sscrofa11.1.dna.toplevel.fa.gz
$ gunzip Sus_scrofa.Sscrofa11.1.105.gtf.gz
```

O primeiro passo do baqcomHisat2.R é indexar o genoma de referência. Essa etapa cria vários arquivos em um diretório específico que serão usados como índice para o mapeamento, que é iniciado logo após a indexação. As opções utilizadas são apresentadas abaixo:

- **-t** => Arquivo fasta do genoma do suíno;
- **-g** => Arquivo gtf;
- **-p** => Número de processadores;
- **-s** => Converte os arquivos sam para bam e os classifica.
- **-q** => Número de amostras a serem processadas.

```bash
$ baqcomHisat2.R –t Sus_scrofa.Sscrofa11.1.dna.toplevel.fa –g Sus_scrofa.Sscrofa11.1.107.gtf -p 16 -s -q 4
## Para mais informações acesse a ajuda usando a tag -h
$ baqcomHisat2.R -h
```

O baqcomHisat2.R automaticamente cria um diretório denominado 02-MappedReadsHISAT2 contendo os arquivos SAM e BAM (pig_unsorted_01.sam, pig_unsorted_01.bam, pig_unsorted_02.sam e pig_unsor-

ted_02.bam). Baixe e instale o Samtools (http://www.sthda.com/english/wiki/install-samtools-on-unix-system) e descompacte o arquivo tar SAMtools. Depois, é necessário indexar a saída dos alinhadores usando o programa Samtools.

```bash
$ tar xvjf samtools-1.15.1.tar.bz2
 cd samtools-1.15.195
 make
 cd
```

Crie o index

A saída do index é em formato bai.

```bash
$ samtools index pig_01_sorted.bam
$ samtools index pig_02_sorted.bam
```

Os arquivos gerados no comando acima (pig_01_sorted.bam e pig_02_sorted.bam) são usados como arquivos de entrada para o StringTie.

##Montagem – StringTie

#Instale o StringTie

Para instalar o StringTie, baixe o pacote na versão 2.2.1 em http://ccb.jhu.edu/software/stringtie, descompacte o arquivo.tar.gz do StringTie e acesse o diretório descompactado.

```bash
$ wget http://ccb.jhu.edu/software/stringtie/dl/stringtie-2.2.1.Linux_x86_64.tar.gz
$ tar xvzf stringtie-2.2.1.Linux_x86_64.tar.gz
$ cd stringtie-2.2.1
$ make release
```

Certifique-se que você esteja na pasta que tenha os programas e os arquivos necessários para iniciar a montagem. Nosso exemplo é com duas amostras, mas cada usuário deve executar os comandos de acordo com seus dados. As opções utilizadas são apresentadas abaixo:

- **-G** => Arquivo de anotação de referência (no formato gtf ou gff3) para orientar o processo de montagem;
- **--rf** => Biblioteca de cadeias fr-firststrand;
- **-o** => Nome do arquivo GTF de saída do StringTie;
- **-A** => Abundâncias de genes serão relatadas no arquivo de saída com o nome fornecido.

```
‘‘‘bash‘‘‘
$ Stringtie 02-MappedReadsHISAT2/pig_01_sorted.bam –G Sus_scrofa.
Sscrofa11.1.107.gtf --rf -o pig_01.gtf -A pig_01.tab
$ Stringtie 02-MappedReadsHISAT2/pig_02_sorted.bam –G Sus_scrofa.
Sscrofa11.1.107.gtf --rf -o pig_02.gtf -A pig_02.tab
#Para mais informações acesse a ajuda usando StringTie -h
```

Saída do StringTie

Ao fim da montagem são gerados arquivos gtf para cada amostra (Figura 4 e 5). Cada arquivo é composto por nove colunas com as seguintes informações:

- **Coluna 1:** Demonstra que a transcrição montada está no cromossomo 1.
- **Coluna 2:** Nome do programa utilizado.
- **Coluna 3:** Transcrição montada.
- **Coluna 4:** Posição inicial do transcrito.
- **Coluna 5:** Posição final do transcrito.

- **Coluna 6:** Pontuação de confiança para a transcrição montada. Atualmente, esse campo não é usado e o StringTie relata um valor constante de 1000 se a transcrição tiver uma conexão com um pacote de alinhamento de leitura.
- **Coluna 7:** Transcrição na fita sense, '+', ou na fita reversa, '-'.
- **Coluna 8:** Região de codificação. O StringTie não usa este campo e simplesmente registra um ".".
- **Coluna 9:** Uma lista separada por ponto e vírgula (;), fornecendo informações adicionais sobre cada transcrição.

```
1 StringTie transcript 1      4191      1000  +  .
1 StringTie exon       1       961      1000  +  .
1 StringTie exon       2371   2465      1000  +  .
1 StringTie exon       3118   4191      1000  +  .
```

Figura 4. Exemplo do arquivo de saída do software StringTie.

A Tabela 1 é um exemplo da saída do StringTie, vale lembrar que a saída do arquivo não vem com as identificações das colunas como na tabela.

Tabela 1. Saída do StringTie.

Coluna 1	Coluna 2	Coluna 3	Coluna 4	Coluna 5	Coluna 6	Coluna 7	Coluna 8
1	StringTie	transcript	1	4.191	1000	+	.
1	StringTie	exon	1	961	1000	+	.
1	StringTie	exon	2.371	2.465	1000	+	.
1	StringTie	exon	3.18	4.191	1000	+	.

```
gene_id "STRG.1"; transcript_id "STRG.1.1"; cov "7.199724"; FPKM "1.244922"; TPM "4.754333";
gene_id "STRG.1"; transcript_id "STRG.1.1"; exon_number "1"; cov "4.964187";
gene_id "STRG.1"; transcript_id "STRG.1.1"; exon_number "2"; cov "19.318583";
gene_id "STRG.1"; transcript_id "STRG.1.1"; exon_number "3"; cov "8.128084";
```

Figura 5. Nona coluna do arquivo de saída do StringTie.

- **gene_id:** Um identificador exclusivo para o gene e éxons com base no nome do arquivo do alinhamento.

- **transcript_id:** Um identificador exclusivo para uma única transcrição e éxons com base no nome do arquivo do alinhamento.
- **exon_number:** Um identificador exclusivo para um único éxon, começando em 1, dentro de uma determinada transcrição.
- **cov:** A cobertura média por base para o transcrito ou éxon.
- **FPKM:** Fragmentos por kilobase de transcrição por milhão de pares lidos. Número de pares de reads alinhados a esse recurso, normalizado pelo número total de fragmentos sequenciados (em milhões) e o comprimento da transcrição (em quilobases).
- **TPM:** Transcrições por milhão. Número de transcritos desse gene específico normalizado primeiro pelo comprimento do gene e depois pela profundidade de sequenciamento.

A Tabela 2 é uma representação da saída da nona coluna do StringTie. Lembre-se que a saída do arquivo não vem com as identificações como na tabela.

Tabela 2. Saída nona coluna do StringTie.

gene_id	trancript_id	exon-number	cov	FPKM	TPM
STRG.1	STRG.1.1		7.199724	1.244922	4.754333
STRG.1	STRG.1.1	1	4.964187		
STRG.1	STRG.1.1	2	19.318583		
STRG.1	STRG.1.1	3	8.128084		

Filtragem Script Perl removendo transcritos com FPKM menor que 0.5

Para maior rigor da predição dos lncRNAs, são removidos transcritos com FPKM menor que 0.5. Este comando é realizado em cada arquivo gtf. Outro comando pode ser usado de acordo com o usuário.

```bash
$ perl -ne '$_ =~ /FPKM "(\d+)/; if ($1>=0.5){ print $_}' pig_01.gtf > Sample_01.gtf
$ perl -ne '$_ =~ /FPKM "(\d+)/; if ($1>=0.5){ print $_}' pig_02.gtf > Sample_02.gtf
```

Mesclagem dos arquivos

Após a filtragem, é realizada a mesclagem das amostras (Sample_01.gtf e Sample_02.gtf) com a ferramenta *merge* do software StringTie. O merge do StringTie toma como entrada uma lista de arquivos GTF e une essas transcrições em um conjunto não redundante de transcrições. Esse modo é usado na análise diferencial para gerar um conjunto global e unificado de transcrições (isoformas) em várias amostras de RNA-Seq. O conjunto de opções e parâmetros utilizado é apresentado abaixo. Em particular, a opção -G (anotação de referência) faz com que o StringTie monte os transfrags dos arquivos GTF de entrada com as transcrições de referência.

- **-G** => Anotação de referência.
- ***gtf** => Irá pegar todos os arquivos gtfs na pasta em que a análise está sendo executada.
- **-o** => Nome do arquivo de saída.

```bash
$ stringtie --merge -G Sus_scrofa.Sscrofa11.1.107.gtf *.gtf -o mesclado.gtf
```

##gffcompare

O gffcompare faz a comparação dos transcritos do StringTie com transcrições conhecidas, gerando os códigos das classes dos transcritos. Estas classes relatam os potenciais transcritos codificantes e não codificantes (Tabela 3).

Tabela 3. Códigos das classes dos transcritos.

Código das Classes	Relação com transcrição de referência
=	Correspondência completa exata de cadeia de íntrons
c	Contido na transcrição de referência
k	Contém transcrição de referência
m	Correspondência completa da cadeia de íntrons em todos os lugares
n	Sobrepõe completamente o íntron da transcrição de referência
j	Possível nova isoforma com pelo menos uma correspondência de junção
e	Éxon único que cobre parcialmente um íntron de referência
o	Fita se sobrepõe com éxons de referência
s	Correspondência de íntrons na fita oposta
x	Antisense
i	Intrônicos
y	Contém uma referência em sua introdução
p	Possível execução da polimerase
r	Pelo menos 50% das bases são soft-masked
u	Intergênicos desconhecidos

Adaptado de: Pertea G and Pertea M. GFF Utilities: GffRead and GffCompare, 2020. DOI: 10.12688/f1000research.23297.1

Baixe o gffcompare em https://ccb.jhu.edu/software/stringtie/gffcompare.shtml e descompacte o arquivo tar.

```
‘‘‘bash‘‘‘
$ wget http://ccb.jhu.edu/software/stringtie/dl/gffcompare-0.12.6.Linux_x86_64.tar.gz
$ tar xvzf gffcompare-0.12.6.tar.gz
```

Depois, execute a ferramenta gffcompare. As opções disponíveis são apresentadas abaixo.

- **-r** => Arquivo GTF da anotação de referência.
- **-G** => Arquivo mesclado.
- **-o** => Nome da saída dos arquivos.

```bash
$ gffcompare -r Sus_scrofa.Sscrofa11.1.107.gtf -G -o mesclado.gtf
```

A saída do gffcompare é composta por seis arquivos (gffcompare.annotated.gtf, gffcompare.loci, gffcompare.stats, gffcompare.tracking, gffcompare.stringtie_asm.gtf.refmap e gffcompare.stringtie_asm.gtf.tmap). O arquivo gffcompare.status mostra o resumo de dados, relatando a sensibilidade, precisão, os novos éxons, íntrons e *loci*. A alta sensibilidade significa que quase todas as transcrições do StringTie correspondem às transcrições conhecidas, ou seja, poucos falsos negativos. Se a precisão é muito menor, indica que muitas das transcrições do StringTie não estão na lista de transcrições conhecidas, que são falsos positivos ou transcrições verdadeiramente de novo. Os novos éxons, íntrons e *loci* indicam quantos dos sítios não foram encontrados na lista de transcrições conhecidas. O arquivo gffcompare.annotated.gtf mostra os códigos das classes.

Contar os códigos das classes que foram formadas

```bash
$ grep -oP "class_code \S+;" gffcompare.annotated.gtf | sort | uniq -c
```

Filtragem removendo as classes codificantes

Essa filtragem remove as classes que não são de interesse, permanecendo somente as classes i, j, o, u, x. O comando grep é indicado para remover as classes que não são desejáveis.

```bash
$ grep -v 'class_code "="' merged.gtf > filtrado.gtf
$ grep -v 'class_code "s"'filtrado.gtf > filtrado1.gt
$ grep –v 'class_code "p"'filtrado1.gtf > filtrado2.gtf
```

Filtragem Script R removendo transcritos com número de exon menor que dois

Para maior confiança da predição dos lncRNAs, os transcritos com éxon único são removidos por meio do script R.

Primeiramente, instale os pacotes do R.

```
“‘ R “‘

install.packages("dplyr")
if (!require("BiocManager", quietly = TRUE)){
install.packages("BiocManager")
BiocManager::install("rtracklayer")

}
```

Em seguida, carregue os pacotes necessários.

```
“‘ R “‘

library(dplyr)
library(rtracklayer)
```

E, então, importe o arquivo gtf.

```
“‘ R “‘

gtf <- import('filtrado2.gtf')
gtf_df <-as.data.frame(gtf)
```

Salve em um objeto o número de genes com um éxon.

```
“‘ R “‘

gtf_1exon <- gtf_df %>%
  count(gene_id,transcript_id,type, name="Exon_count") %>%
  filter(type=='exon' & Exon_count == 1) %>%
  select(-type)
```

Salve também em um objeto o número de genes com dois ou mais éxons.

```
“‘ R “‘

gtf_2oumais <- gtf_df %>%
  count(gene_id,transcript_id,type, name="Exon_count") %>%
  filter(type=='exon' & Exon_count >= 2) %>%
  select(-type)
```

Por fim, crie um objeto para filtrar com os dados.

```
“‘ R “‘

gtf_filtered <- gtf_df %>%
  dplyr::filter(transcript_id %in% gtf_2oumais$transcript_id)
```

E salve o arquivo gtf exportado.

```
“‘ R “‘

export(gtf_filtered, "gtf_filtered.gtf")
```

##gffread

Após remover os transcritos com exon único, o arquivo gtf é transformado em um arquivo fasta com o programa gffread. Instale o gffread e depois execute o programa.

```
““‘ bash “‘

$ git clone https://github.com/gpertea/gffread
$ cd gffread
$ make release
```

O gffread realiza a conversão de arquivo gtf em fasta.

- **-w** => Nome do arquivo de saída (fasta);
- **-g** => Arquivo do genoma do suíno no formato fasta.

```bash
$ gffread -w result.fa -g Sus_scrofa.Sscrofa11.1.dna.toplevel.fa gtf_filtered.gtf
```

Ao fim da execução da etapa anterior, a saída do gffread (result.fa) é filtrada pelo tamanho dos transcritos pela linha de comando abaixo, permanecendo na análise os transcritos com tamanho ≥200 nucleotídeos.

Filtragem pelo tamanho dos transcritos

Remova todos os transcritos com tamanho menor que 200 nucleotídeos usando o comando abaixo:

```bash
$ perl -ne 'chomp; if (/>/){ print "\n$_\t"}else{ print $_}' result.fa | perl -ne '/^(.+)\t(.+)/; if (length($2)>=200){ print "$1\t" . length($2) . "\n$2\n";}' > resultado_final.fa
```

Potencial de codificação

Faça a predição do potencial de codificação dos transcritos com os programas CPC2, CNCI, CPAT e PLEK. Os parâmetros utilizados neste documento foram definidos de acordo com os dados utilizados.

CPC2

O CPC2 é um software para avaliar o potencial de codificação de proteínas dos transcritos. Para instalá-lo é necessário o pacote Biopython. Uma versão local pode ser baixada em (http://biopython.org/wiki/Download).

```bash
$ CPC2_standalone-1.0.1/
$ cd CPC2-beta
$ export CPC_HOME="$PWD"
$ cd libs/libsvm
$ gzip -dc libsvm-3.18.tar.gz | tar xf –
$ cd libsvm-3.18
$ make clean && make
```

Execute o CPC2.py utilizando os parâmetros apresentados abaixo:

- **-i** => Arquivo fasta gerado após a montagem e os filtros;
- **-o** => Nome do arquivo de saída;
- **--ORF** => Saída da posição inicial da ORF mais longa.

```bash
$ python3 CPC2_standalone-1.0.1/bin/CPC2.py -i resultado_final.fa -o CPC2.txt --ORF
```

Após o fim do passo anterior, execute o seguinte comando para continuar realizando a análise somente com os potenciais transcritos não codificantes.

```bash
$ cat CPC2_txt | grep noncoding > noncoding_CPC2
```

CNCI

O CNCI identifica sequências codificadoras e não codificantes de proteínas, sendo uma ferramenta de classificação precisa de transcrições de dados de sequenciamento.

Instale o CNCI e execute-o em python.

```bash
$ git clone git@github.com:www-bioinfo-org/CNCI.git
$ cd CNCI
$ unzip libsvm-3.0.zip
$ cd libsvm-3.0
$ make
$ cd ..
```

Primeiramente, é necessário rodar o *compare.py*, para depois executar o CNCI. Para o compare são necessários os arquivos gtf codificante e não codificante, que são oriundos do arquivo gtf do genoma do suíno (Sus_scrofa.Sscrofa11.1.107.gtf), baixado no início da análise. É só realizar a separação dos codificantes e não codificantes com o auxílio do comando grep.

```bash
$ grep 'gene_biotype "protein_coding"' Sus_scrofa.Sscrofa11.1.107.gtf > coding.gtf
$ grep -v 'gene_biotype "protein_coding"'Sus_scrofa.Sscrofa11.1.107.gtf > ncrna.gtf
```

Importante: O cabeçalho dos arquivos coding.gtf e ncrna.gtf não devem apresentar ##, pois com essa informação o comando dará erro.

O *compare.py* compara as transcrições montadas com a anotação referência. É uma etapa obrigatória para realizar o CNCI.py. Os arquivos definidos pelos parâmetros –n, -c e –i devem possuir as informações gene_id e transcript_id.

- **-n** => RNAs não codificantes.
- **-c** => Proteínas codificantes.
- **-i** => Arquivo de entrada.
- **-o** => Arquivo de saída dos resultados.

```bash
$ python3 /CNCI/compare.py -c coding.gtf -n ncrna.gtf -i gtf_filtered.gtf
-o result_compare
```

A saída do compare gera o arquivo potentially_novel.gtf, que é usado na próxima etapa. Para usar o CNCI.py é necessário transformar o arquivo fasta do genoma em formato dois bits, com o comando:

```bash
$ faToTwoBit Sus_scrofa.Sscrofa11.1.dna.toplevel.fa genome_pig.2bit
```

Execute o CNCI.py com os seguintes parâmetros:

- **-f** => Arquivo de entrada (gerado pelo compare.py);
- **-g** => Parâmetro obrigatório ao utilizar arquivos no formato gtf;
- **-m** => Atribui os modelos de classificação (ve - espécies de vertebrados);
- **-p** => Número de processadores;
- **-d** => Genoma do suíno convertido 2bit;
- **-o** => Diretório de saída dos resultados.

```bash
$ python3 /CNCI/CNCI.py -f potentially_novel.gtf -g -m ve -p 8 -d geno-
me_pig.2bit –o CNCI_codificacao
```

CPAT

A ferramenta CPAT reconhece rapidamente transcrições codificantes e não codificantes de um grande grupo de candidatos. Ele deve ser instalado com python3.5 ou versões posteriores.

```bash
$ pip3 instalar git+https://github.com/liguowang/cpat.git
```

O programa fornece o modelo logit pré-construído e a tabela hexâmero para humanos, ratos, *zebrafish* e moscas. Para outras espécies (suínos, por exemplo), é necessário construir os modelos.

A ferramenta make_hexamer_tab.py calcula a frequência do hexâmero das sequências codificante e não codificante. Para esse cálculo, baixe os arquivos fasta em https://www.ensembl.org/info/data/ftp/index.html e descompacte.

```bash
##Sequencia CDS
$ wget http://ftp.ensembl.org/pub/release-107/fasta/sus_scrofa/cds/
Sus_scrofa.Sscrofa11.1.cds.all.fa.gz
$ gunzip Sus_scrofa.Sscrofa11.1.cds.all.fa.gz
##Sequencia ncrna
$ wget http://ftp.ensembl.org/pub/release-107/fasta/sus_scrofa/ncrna/
Sus_scrofa.Sscrofa11.1.ncrna.fa.gz
$ gunzip Sus_scrofa.Sscrofa11.1.cds.all.fa.gz
```

Para utilizar o hexamer_tab.py, os seguintes parâmetros devem ser especificados:

- **-c** => Arquivo das sequências codificantes.
- **-n** => Arquivo das sequências não codificantes.

```bash
make_hexamer_tab.py -c Sus_scrofa.Sscrofa11.1.cds.all.fa –n Sus_
scrofa.Sscrofa11.1.ncrna.fa > Sscrofa_Hexamer.tsv
```

Para a construção do modelo de regressão logística, baixe e descompacte o arquivo de cdna, seguindo as instruções do passo anterior, e crie o modelo. Os seguintes parâmetros devem ser especificados:

- **-x** => Tabela de frequências de hexâmetros construída na etapa anterior.
- **-c** => Arquivo de sequências de RNAs codificantes de proteína no formato FASTA.
- **-n** => Arquivo de sequências de RNAs não codificantes no formato FASTA.
- **-o** => Prefixo do arquivo de saída.

```bash
$ make_logitModel.py -x Sscrofa_Hexamer.tsv -c Sus_scrofa.Sscrofa11.1.cdna.all.fa -n Sus_scrofa.Sscrofa11.1.ncrna.fa -o modelo
```

Segundo Wang et al. (2013), o CPAT dará exatamente os mesmos resultados que o NCBI ORFfinder para identificar as ORFs. Execute o CPAT especificando os seguintes parâmetros:

- **-x** => Tabela de frequências de hexâmetros;
- **-d** => Modelo de Logit;
- **--antisense** => Quando essa opção não é especificada, só buscará ORFs da cadeia sense;
- **--top-orf** => Número para relatar todas as ORFs;
- **-g** => Arquivo fasta gerado após a montagem e filtragem;
- **-o** => Nome do arquivo de saída.

```bash
$ cpat.py -x Sscrofa_Hexamer.tsv -d modelo.logit.RData --antisense --top-orf=100 -g resultado_final.fa -o cod_cpat
```

Ao fim da execução são gerados os seguintes arquivos:

- **cod_capt.ORF_seqs.fa:** As principais sequências.

- **cod_capt _prob.tsv:** Informações sobre a ORF (cadeia, quadro, início, fim, tamanho, pontuação Fickett, pontuação Hexamer) e probabilidade de codificação).
- **cod_capt _prob.best.tsv:** A informação da melhor ORF. Este arquivo é um subconjunto de "cod_capt.ORF_prob.tsv".
- **cod_capt _ORF.txt:** IDs de sequências sem ORF encontrada.
- **cod_capt.r:** Arquivo Rscript.
- **CPAT_run_info.log:** Arquivo de log.

Para separar os transcritos codificantes dos não codificantes, o limite de probabilidade de codificação humana (CP) foi utilizado, pois é o menor limite entre as espécies para maior confiança dos dados (CP ≥0,364 indica sequência de codificação, CP < 0,364 indica sequência não codificante). O arquivo co_cpat.ORF_prob.best.tsv apresenta as informações das ORFs, como: cadeia, quadro, início, fim, tamanho, pontuação Fickett, pontuação Hexamer e probabilidade de codificação. Essa última é localizada na décima primeira coluna. Com o comando awk, filtre os transcritos conforme descrito abaixo para que somente os potenciais não codificantes permaneçam na análise.

```bash
$ awk '{if($11<0.364){print $_}}' co_cpat.ORF_prob.best.tsv > CPAT_noncoding
```

#PLEK

Em conjuntos de dados de vertebrados, o programa pode ser executado com as configurações padrão. Baixe em https://sourceforge.net/projects/plek/files/ e descompacte.

```bash
$ tar zvxf PLEK.1.2.tar.gz
$ cd PLEK.1.2
$ python3 PLEK_setup.py
```

PLEK.py é usado para diferenciar lncRNAs de mRNAs. Execute-o, especificando os seguintes parâmetros:

- **-fasta** => Arquivo no formato fasta gerado após a montagem e as filtragens;
- **-out** => Nome do arquivo de saída;
- **-thread** => Números de processadores;
- **-minlength** => Comprimento mínimo das sequências.

```bash
$ python3 PLEK.1.2/PLEK.py -fasta resultado_final.fa -out predicted
-thread 10 -minlength 200
```

##Filtra os transcritos com o comando grep, mantendo somente os potenciais não codificantes.

```bash
$ cat predicted | grep Non-coding > PLEK_non-coding.txt
```

#Diagrama de Venn

Identifica os transcritos em comum entre as quatro ferramentas de predição de codificação utilizadas (CPC2, CNCI, PLEK e CPAT).

O diagrama pode ser construído pelo R ou online em http://www.interactivenn.net/.

##Pfam

É um banco de dados baseado em perfis de modelos ocultos de Markov (HMMs), que combina alta qualidade e integridade. O Pfam consiste nas partes A e B. O Pfam-A é acurado e contém famílias de domínios de proteínas bem caracterizadas com alinhamentos de alta qualidade, que são mantidos usando alinhamentos verificados manualmente e HMMs para localizar e alinhar todos os membros. Pfam-B contém famílias de sequências que foram

geradas automaticamente pela aplicação do algoritmo Domainer para agrupar e alinhar as sequências de proteínas remanescentes após a remoção dos domínios Pfam-A (Sonnhammer; Eddy; Durbin, 1997).

Instale e descompacte

```bash
$ wget ftp://ftp.ebi.ac.uk/pub/databases/Pfam/current_release/Pfam-A.hmm.gz
$ gunzip Pfam-A.hmm.gz
```

Primeiramente, execute o hmmpress, passo necessário para que o hmmscan funcione. Essa etapa prepara um banco de dados, construindo arquivos binários compactados e indexados para o hmmscan.

```bash
$ hmmpress Pfam-A.hmm
```

Quatro arquivos binários são criados: Pfam-A.hmm.h3f, Pfam-A.hmm.h3i, Pfam-A.hmm.h3m e Pfam-A.hmm.h3p.

Agora analise as sequências usando seu banco de dados, executando o Pfam-A.hmm.h3p com os seguintes parâmetros:

- **--cpu** => Número de processadores.
- **--domtblot** => Salva arquivos em formato tabular simples.
- **-E** => Número de falsos positivos relatados por consulta.
- **--domE** => Número esperado de domínios falsos positivos.
- **Pfam-A.hmm** => Arquivo baixado na etapa anterior.
- **Transcritos.fasta** => Arquivo fasta gerado a partir dos transcritos em comum com as ferramentas de codificação.

```
“‘bash“‘
$ hmmscan --cpu 10 --domtblout PFAM_filtered.out -E 0.001 --domE
0.001 Pfam-A.hmm transcritos.fasta > pfam_filtered.log
```

Após a identificação dos potenciais lncRNAs, a análise de expressão diferencial e a caracterização dos transcritos podem ser realizadas com software R.

A análise de expressão gênica pode ser feita pelo DESeq2, edgeR ou limma, que são pacotes do Bioconductor para análise de expressão diferencial, que recebem como entrada uma matriz de contagens de leitura.

O script Python prepDE.py, baixado em http://ccb.jhu.edu/software/stringtie/dl/prepDE.py, usa a fórmula *reads_per_transcript = cobertura * transcript_len / read_len* para determinar a contagem. É usado para extrair essas informações de contagem de leituras diretamente dos arquivos gerados por StringTie (executado com o parâmetro -e). Execute-o, especificando o seguinte parâmetro:

```
“‘bash“‘
$ python3 prepDE.py -i dados.txt
```

A saída do predDE.py gera dois arquivos CSV, que contêm as matrizes de contagem para genes e transcrições. Posteriormente, realize a análise de expressão diferencial.

Considerações finais

Os papéis regulatórios dos lncRNAs ainda não foram totalmente explorados. É extremamente importante entender como funcionam as ferramentas de bioinformática para identificá-los. Aqui, descrevemos um material em português com o passo a passo para a identificação dos potenciais lncRNAs, com intuito de facilitar e maximizar a realização dessas análises e, assim,

aumentar a base de dados e o conhecimento em relação ao funcionamento dos mesmos.

Desta forma, o objetivo deste documento é fornecer informações para as análises de lncRNAs, em português, de forma detalhada e prática, tendo como público alvo estudantes de graduação, pós-graduação e cientistas que pretendam utilizar estas abordagens de análises de expressão gênica em suínos ou outras espécies de animais. A disponibilização de materiais em português possibilita um aumento na igualdade de acesso à educação de qualidade, garantindo uma melhor qualificação profissional e permitindo a aquisição de habilidades relevantes em análises transcriptômicas. Trabalhos como este, estão alinhados à Meta 4.7 do ODS 4, que visa garantir que todos os alunos adquiram conhecimentos e habilidades necessárias para promover o desenvolvimento sustentável, inclusive, entre outros, por meio da educação para o desenvolvimento sustentável. Para isso, são apresentadas as etapas para realização dessa análise e as ferramentas empregadas em cada uma dessas etapas.

Agradecimentos

Ao Instituto Nacional de Ciência e Tecnologia/Ciência Animal (INCT/CA, Processo CNPq Processo 465377/2014-9) da Universidade Federal de Viçosa (UFV). A Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES)/PROEX Processo 88887.844747/2023-00 pelo financiamento do projeto, pelo apoio ao projeto (Código de Financiamento 001) e pela bolsa de FGC. Ao CNPq pela bolsa de produtividade de MCL, AMGI e SEFG, pela bolsa de pós-doutorado de HCO e de doutorado de KAC e SAT.

Referências

BOLGER, A. M.; LOHSE, M.; USADEL, B. Trimmomatic: a flexible trimmer for Illumina sequence data. **Bioinformatics**, v. 30, n. 15, p. 2114-2120, 2014.

DOZMOROV, M. G.; GILES, C. B.; KOELSCH, K. A.; WREN, J. D. Systematic classification of non-coding RNAs by epigenomic similarity. **BMC Bioinformatics**, v. 14, n. 10, p. 1-12, 2013.

ESTEVE-CODINA, A.; KOFLER, R.; PALMIERI, N.; BUSSOTTI, G.; NOTREDAME, C.; PÉREZ-ENCISO, M. Exploring the gonad transcriptome of two extreme male pigs with RNA-seq. **BMC Genomics**, v. 12, n. 1, p. 1-14, 2011.

HERBERT, Z. T.; KERSHNER, J. P.; BUTTY, V. L. et al. Cross-site comparison of ribosomal depletion kits for Illumina RNAseq library construction. **BMC Genomics**, v. 19, n. 1, p. 1-10, 2018.

HE, Y.; VOGELSTEIN, B.; VELCULESCU, V. E.; PAPADOPOULOS, N.; KINZLER, K. W. The antisense transcriptomes of human cells. Science, v. 322, n. 5909, p. 1855-1857, 2008.

HUANG, W.; ZHANG, X.; LI, A.; XIE, L.; MIAO, X. Genome-wide analysis of mRNAs and lncRNAs of intramuscular fat related to lipid metabolism in two pig breeds. **Cellular Physiology and Biochemistry**, v. 50, n. 6, p. 2406-2422, 2018.

JOHNSSON, P.; LIPOVICH, L.; GRANDÉR, D.; MORRIS, K. V. Evolutionary conservation of long non-coding RNAs; sequence, structure, function. **Biochimica et Biophysica Acta (BBA)-General Subjects**, v. 1840, n. 3, p. 1063-1071, 2014.

KANG, Y. J.; YANG, D. C.; KONG, L.; HOU, M.; MENG, Y. Q.; WEI, L.; GAO, G. CPC2: a fast and accurate coding potential calculator based on sequence intrinsic features. **Nucleic Acids Research**, v. 45, n. W1, p. W12-W16, 2017.

KIM, D.; LANGMEAD, B.; SALZBERG, S. L. HISAT: a fast spliced aligner with low memory requirements. **Nature Methods**, v.12, n.4, p. 357-360, 2015.

KONG, L.; ZHANG, Y.; YE, Z.Q.; LIU, X. Q.; ZHAO, S. Q.; WEI, L.; GAO, G. CPC: assess the protein-coding potential of transcripts using sequence features and support vector machine. **Nucleic Acids Research**, v. 35, n. suppl_2, p. W345-W349, 2007.

LI, A.; ZHANG, J.; ZHOU, Z. PLEK: a tool for predicting long non-coding RNAs and messenger RNAs based on an improved k-mer scheme. **BMC Bioinformatics**, v. 15, n. 1, p. 1-10, 2014.

LI, J.; LIU, C. Coding or noncoding, the converging concepts of RNAs. **Frontiers in Genetics**, v. 10, p. 496, 2019.

Li, J.; Zhang, X.; Liu, C. The computational approaches of lncRNA identification based on coding potential: status quo and challenges. **Computational and Structural Biotechnology Journal**, v. 18, p. 3666-3677, 2020.

LIU, X.; ZHAO, J.; XUE, L.; ZHAO, T.; DING, W.; HAN, Y.; YE, H. A comparison of transcriptome analysis methods with reference genome. **BMC Genomics**, v. 23, n. 1, p. 1-15, 2022.

MARÇAIS, G.; KINGSFORD, C. A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. **Bioinformatics**, v. 27, n. 6, p. 764-770, 2011.

MARTIN, M. Cutadapt removes adapter sequences from high-throughput sequencing reads. **EMBnet. journal**, v. 17, n. 1, p. 10-12, 2011.

MATTICK, J. S.; RINN, J. L. Discovery and annotation of long noncoding rnas. **Nature Structural & Molecular Biology**, v. 22, n. 1, p. 5-7, 2015.

MISTRY, J.; CHUGURANSKY, S.; WILLIAMS, L. et al. Pfam: The protein families database in 2021. **Nucleic Acids Research**, v. 49, n. D1, p. D412-D419, 2021.

PERTEA, M.; PERTEA, G. M.; ANTONESCU, C. M.; CHANG, T. C.; MENDELL, J. T.; SALZBERG, S. L. StringTie enables improved reconstruction of a transcriptome from RNA-seq reads. **Nature Biotechnology**, v. 33, n. 3, p. 290-295, 2015.

RAMIREZ-GONZALEZ, R. H.; BONNAL, R.; CACCAMO, M.; MACLEAN, D. Bio-samtools: Ruby bindings para SAMtools, uma biblioteca para acessar arquivos BAM contendo alinhamentos de sequência de alto rendimento. **Código-Fonte para Biologia e Medicina**, v. 7, n. 1, p. 1-6, 2012.

R Core Team (2021). **R: A language and environment for statistical computing**. R Foundation for Statistical Computing, Vienna, Austria, 2021. Disponível em: https://www.R-project.org/. Acesso em: 9 out. 2023.

RINN, J. L.; KERTESZ, M.; WANG, J. K. et al. Functional demarcation of active and silent chromatin domains in human HOX loci by noncoding RNAs. **Cell**, v. 129, n. 7, p. 1311-1323, 2007.

SONNHAMMER, E. L.; EDDY, S. R.; DURBIN, R. Pfam: a comprehensive database of protein domain families based on seed alignments. **Proteins: Structure, Function, and Bioinformatics**, v. 28, n. 3, p. 405-420, 1997.

SLEUTELS, F.; ZWART, R.; BARLOW, D. P. The non-coding Air RNA is required for silencing autosomal imprinted genes. **Nature**, v. 415, n. 6873, p. 810-813, 2002.

SUN, L; LUO, H.; BU, D. et al. Utilizing sequence intrinsic composition to classify protein-coding and long non-coding transcripts. **Nucleic Acids Research**, v. 41, n. 17, p. e166, 2013.

TAFT, R. J.; PANG, K. C.; MERCER, T. R.; DINGER, M.; MATTICK, J. S. Non-coding RNAs: regulators of disease. **The Journal of Pathology**: A Journal of the Pathological Society of Great Britain and Ireland, v. 220, n. 2, p. 126-139, 2010.

VECERA, M.; SANA, J.; OPPELT, J.; TICHY, B.; ALENA, K.; LIPINA, R.; SMRCKA, M.; JANCALEK, R.; HERMANOVA, M.; KREN, L.; SLABY, O. Testing of library preparation methods for transcriptome sequencing of real life glioblastoma and brain tissue specimens: A comparative study with special focus on long non-coding RNAs. **PLoS One**, v. 14, n. 2, p. e0211978, 2019.

WANG, H.; IACOANGELI, A.; LIN; D.; WILLIAMS, K.;; DENMAN, R. B.; HELLEN, C. U.; TIEDGE, H. Dendritic BC1 RNA in translational control mechanisms. **The Journal of Cell Biology**, v. 171, n. 5, p. 811-821, 2005.

WANG, L.; PARK, H. J.; DASARI, S.; WANG, S.; KOCHER, J. P.; LI, W. CPAT: Coding-Potential Assessment Tool using an alignment-free logistic regression model. **Nucleic Acids Research**, v. 41, n. 6, p. e74-e74, 2013.

WANG, X.; ARAI, S.; SONG, X. et al. Induced ncRNAs allosterically modify RNA-binding proteins in cis to inhibit transcription. **Nature**, v. 454, n. 7200, p. 126-130, 2008.

WITTKOPP, P. J.; HAERUM, B. K.; CLARK, A. G. Evolutionary changes in cis and trans gene regulation. **Nature**, v. 430, n. 6995, p. 85-88, 2004.

YAN, M. D.; HONG, C. C.; LAI, G. M.; CHENG, A. L.; LIN, Y. W.; CHUANG, S. E. Identification and characterization of a novel gene Saf transcribed from the opposite strand of Fas. **Human Molecular Genetics**, v. 14, n. 11, p. 1465-1474, 2005.

YAN, P.; LUO, S.; LU, J. Y.; SHEN, X. Cis- and trans-acting lncRNAs in pluripotency and reprogramming. **Current Opinion in Genetics & Development**, v. 46, p. 170-178, 2017.

ZHANG, P.; WU, W.; CHEN, Q.; CHEN, M. Non-coding RNAs and their integrated networks. **Journal of Integrative Bioinformatics**, v. 16, n. 3, 2019.

Embrapa
MINISTÉRIO DA AGRICULTURA E PECUÁRIA
GOVERNO FEDERAL
BRASIL
UNIÃO E RECONSTRUÇÃO