ngfftools manual

version 202204

The main version of this document can be found at https://github.com/dengcao3/ngfftools.

Contents

1. Introduction

Program:
ngfftools - tools for processing Next-generation Genomic Feature Format (ngff) files

Version:
1.00 2021-08-03
1.01 2022-10-01

Usage:
ngfftools [module_name] [Options]
ngfftools [module_name] -help for detail parameter information of specific module

Modules:
format format conversion among ngff, gtf, gff3 format.
grep grep value in 1st-8th column or tags in attribute column.
merge merge multiple annotation files into one file.
extract retrieve child or parent elements of the given id(s).
abs2rel convert absolute position to relative position.
rel2abs convert relative position to absolute position based on one reference ngff.
seq retrieve sequence in annotation file.

Authors:
Deng Cao, concepts, design, algorithm
Huang Ziyan, coding implementation

2. format

Convert between two formats, including NGFF, GTF, and GFF3.

2.1. parameters

ngfftools format <options>

parameter	Type	Default	Alternatives	Description
-I|--infile*	String	-	-	input filename
-IF|--informat	String	-	<ngff/gtf/gff3>	input format
-O|--outfile	String	-	-	output filename
-OF|--outformat	String	-	<ngff/gtf/gff3>	output format, if set --outfile and --outformat is null, the output format is inferred from the suffix of --outfile
-T|--name_type	String	not_ncbi	<ncbi/not_ncbi>	if set 'ncbi' , the product ID will be protein_id (CDS) or product name (miRNA)
-P|--prefix	String	""	-	Prefix of output annotation filename

*: mandatory

2.2. Example

*(1) ngfftools format --infile test.ngff --informat ngff --outfile out.gtf --prefix test_*
(2) cat template.ngff | ngfftools format --infile - --informat ngff --outfile OUT.gtf

The input file can be a file, like the example (1), and you can remove --informat if you have proper suffix in the input file name.

The input file can also be standard input stream, like the example (2), and in this case, the --informat is mandatory.

3. grep

Grep value in 1st-8th column or tags in attrbutes from 9th column. Grepping values in annotation files, especially in the 9th column, is troublesome for Linux commands solution (such as grep, cut). So this grep module is created for conveniently searching value from ngff format files. It allows two modes, match and regular expression.

3.1. parameters

ngfftools grep --grep 'TAG OPERATOR VALUE' input.ngff >output.ngff

'TAG OPERATO RVALUE':

(1) TAG: 1st-8th column (seqId, start, end, strand, level, featureType, id, parentId) and tag in attribute column (9th column)

(2) OPERATOR: (number: >,<,==,<=,<=,!=), (string: ==), (pattern matching: =~, !~)

(3) VALUE: number, "string", /value/(match pattern)

3.2. Example

ngfftools grep --grep 'seqId!~/chrM/ && (id=~/^MSTG/||id=~/^AT/i) && level=="product" && count>=5' --infile input.ngff --informat ngff > grep.ngff

Notice: the value should not contain reserved symbol, such as: **"**, **'**, **[**, **]**, **{**, **}**. If you have attribute like:

Dbxref=GeneID:100287102,HGNC:HGNC:37102; attribute3=multi1,multi2;

You can set the grep as:

--grep 'Dbxref == "GeneID:100287102,HGNC:HGNC:37102"'

--grep 'attribute3 == "multi1,multi2"'

4. merge

merge multiple annotation files into one file. Sometimes, researchers will add annotation results from different methods (such as Iso-seq for high quality genome annotation) to reference annotation file, this function will benefit them.

4.1. parameters

parameter	Type	Description
-L|--list	String	List of input filenames. One file per line. (the first column: file name, the second column: prefix).
-R|--reference	String	reference annotation file. The first ngff file or the first gff3/gtf annotation file (if there's no ngff file in the --list) will be used as backbone.
-OV|--overlap_percent	String	the min overlap percent (overlapped length / the reference locus or sublocus length) when merge two locus. Default is 0.5
-H|--header	String	copy the header in this FILE to output ngff file. By default, header will inherit from the first ngff annotation file

4.2. Example

(1) ngfftools merge --outfile merge.ngff --list merge.list --reference test.ngff --overlap_percent 0.8 1>merge.log 2>merge.err

(2) ngfftools merge --list mergeTwo.list --outfile mergeTwo.ngff &>mergeTwo.error

Following is the content of mergeTwo.list, and the backbone file is *template.ngff*:

*forMerge.gff3 isoseq_*

template.ngff
*forMerge.gtf genoma_*

5. abs2rel

Convert absolute position to relative position (supported level type: processed and product). The absolute position is the position on the genome; and the relative position is the position based on the id itself. For example, the mRNA1 has two CDS, 101-120 (cds1) and 161-191 (cds2) at the positive strand of chr1. The relative position regions for it are 1-20, 21-51, respectively. While if the mRNA is at the negative strand, the relative position regions for it are 32-51, 1-31, respectively.

During the sequence analyses, researchers may need to get the CDS position regions on the protein coding region or the exon position on the mRNA, in these cases, it's available through abs2rel function. Moreover, the relative position is better to describe circular RNA sequence.

5.1. parameters

ngfftools abs2rel [options] <input.ngff> [ouput.ngff]

options:
-C|--abs2rel 'level=relative'
Level can be product or processed or both, use ";" to separate multiple levels, and you can see detail in example.

5.2. Example

(1) ngfftools abs2rel --abs2rel 'product=relative' --infile test.ngff --outfile product_relative.ngff --informat ngff
(2) *ngfftools abs2rel --abs2rel 'processed=relative;product=relative' --infile template.ngff --outfile proc_prod_relative.ngff*

6. rel2abs

Convert relative position to absolute position according to reference ngff. Variants described on the DNA level are mostly reported in relation to a specific gene based on a "coding DNA reference sequence". When a coding DNA reference sequence is used, the description of the variant starts with "c." (for example c.4375C>T). Sometimes researcher wants to check this mutation point based on a "genomic reference sequence", starting with "g." (for example g.3240776G>A). We can use rel2abs function to obtain the absolute position on the genome.

6.1. parameters

ngfftools rel2abs [Options]
Input has two inputs: --infile (reference ngff file) and --RELposFile (relative position ngff file)

parameter:
-RE|--RELposFile relative position ngff file

6.2. Example

relative.ngff:

```
Chr1   -3    -5    +     product mature  L01.t11.p1     L01.t11 product_biotype=miRNA;
Chr1   -1    -3    +     product mature  L01.t11.p2     L01.t11 product_biotype=miRNA;
Chr1   -4    -6    +     product mature  L01.t11.p2     L01.t11 product_biotype=miRNA;
Chr1   -10   -12   -      product CDS    L01.t12.p1     L01.t12 product_biotype=ORF;
```

the simplest relative.ngff (must have: start end level id)

. -1 -3 . product . XP_011287422.1 . .
. -173 -177 . product . XP_011287422.1 . . ***
. -3 -6 . processed . rna3 . .
*** if the relative region cross two CDS/exon, the output will split into two parts.

ngfftools rel2abs --outfile ABSpos.ngff --infile test.ngff --informat ngff --RELposFile relative.ngff

7. seq

Retrieve sequence in annotation file.

7.1. parameters

parameter	Type	Default	Description
-OP|--outpre	String	out	output prefix
-GM |--genomeFile*	String	-	genome sequence file
-S |--species_type*	String	-	species type for choosing codon system(Vertebrate /Yeast/ Protozoan/ Invertebrate/ Ciliate/ Echinoderm...)

This function will produce four types of sequence files:

File	Detail
{outpre}_nucleotide.fasta	non-regulator nucleotide sequence in level locus, primary, processed and product
{outpre}_CDS.fasta	CDS sequence (featureType is CDS)
{outpre}_pep.fasta	Protein sequence translated from {outpre}_CDS.fasta
{outpre}_regulator.fasta	regulator nucleotide sequence in level locus, primary, processed and product

7.2. Example

ngfftools seq --infile test.ngff --informat ngff --genomeFile test.fasta --species_type Vertebrate --outpre TEST >retrieveSequence.log 2>retrieveSequence.err

8. extract

extract child or parent elements of the given id(s). If the retrieving direction is child, the output will extract the child and their off-springs terms recursively. For example, id L01.pt1 is 'primary' level. In processed level, L01.pt1 produces L01.t1 and L01.t2. And in product level, L01.t1 produces L01.t1.p1, L01.t2 produces L01.t2.p1. So in child direction, the output contains ids: L01.pt1, L01.t1, L01.t2, L01.t1.p1, L01.t2.p1. In the same manner, if the extract direction is parent, the output will have the parent and ancestral terms recursively.

8.1. parameters

-U |--unit id:<parent/child>. use "," to separate multiple ids

8.2. Example

ngfftools unit --infile input.ngff --informat ngff --unit L01.t1:parent:L01.t2:child > unit.ngff