IgFamily v0.12.1h

Technical manual v0.1.10

Overview

Features

­­­­­

*For further information on features, refer to associated entries -*

**- FASTA utility tools:**

Read FASTA files into runtime. Custom FASTA creation utility provides functionality for tailored output files. This utlility can be used to create required FASTA format –

>[ACCESSION]|[NAME]|[TYPE]|[SPECIES]|

Accession field is a housekeeping field and is not required for runtime. Name, type, and species fields are used in data structure creation, association, and function.

|  |  |
| --- | --- |
| FASTA parse function | pp. |
| FASTA file utilities | pp. |

**- Peptide file compatibility:**

Read peptide files into runtime. Currently supported are *PEAKS v8.0 DE NOVO* de novo peptides .csv export, *PEAKS v8.0 SPIDER* protein peptide .csv export, and *NOVOR v1.1* de novo peptides .csv export.

*PEAKS v8.0 DE NOVO* de novo peptides peptide file assigns scan number, peptide with modification, and amino acid local confidence score. Note that here *PEAKS* *v8.0 DE NOVO* assigns individual export accessions to replicate peptide assignments.

*PEAKS v8.0 SPIDER* protein peptide file assigns scan number, peptide with modification, spectral count, and -10IgP certainty score.

*NOVOR v1.1* de novo peptides file assigns scan number, peptide with modification, and amino acid local confidence score. Note that here *NOVOR v1.1* assigns individual export accessions to replicate peptide assignments.

|  |  |
| --- | --- |
| Peptide file parse functions | pp. |

**- *msconvert* external integration**

*msconvert* is able to be called through a user defined interface option to convert .wiff and .wiff.scan files into the *Mascot Generic Format* .mgf file type. Various command line options may be selected with peak-picking as the default option. The generated file is created in the same folder as the input file.

|  |  |
| --- | --- |
| *msconvert* convert command line functions | pp. |
|  |  |

**- *NOVOR v1.1* external integration**

*NOVOR v1.1* is able to be called through a user defined interface option to generate *NOVOR v1.1* de novo peptide files in the .csv file type. Various command line options may be selected. The generated file is created in the same folder as the input file.

|  |  |
| --- | --- |
| *NOVOR v1.1* convert command line functions | pp. |

**- Local directory runtime functionality**

The *IgFamily* program is able to be run in local directory or filesystem directory file mode. In local directory file mode the user places the required files in the *IgFamily* root directory and executes the program. The input file and output files are moved to a folder created in the root directory with the name of the input file sample - The input file contains the sample name and is supported in the three input data types.

|  |  |
| --- | --- |
| Local directory runtimefunctions | pp. |

**- Data filesystem**

The *IgFamily* program is able to be run in local directory or filesystem directory file mode. In filesystem mode the program accessions a dedicated file association structure to retrieve and export files. The filesystem is currently defined on the *FATELVIS* network assisted storage device. The user is required to accession a file initially, however there is proposed functionality for dynamic file management with the *ISO/IEC TS 18822:2015* filesystem library.

|  |  |
| --- | --- |
| Data filesystem functions | pp. |

**- Runtime user interface**

On execution of the *IgFamily* program, the user is greeted with an interactive menu. The user is able to access FASTA file utilities, *msconvert* command line tools, *NOVOR v1.1* command line tools, and select program parameters. Program parameters include local or filesystem file modes, FASTA file selection, and peptide assignment selection.

|  |  |
| --- | --- |
| User interface functions | pp. |

**- FASTA data structuring**

Following user selection of runtime parameters, the *IgFamily* program will parse the FASTA file(s). The data are stored on contiguous RAM buffers at runtime. The [NAME], [TYPE], and [SPECIES] fields have runtime functionality, although the [ACCESSION] field is retained for FASTA utility functions. An additional data type is created to define those FASTA accessions that are immunoglobulin.

|  |  |
| --- | --- |
| FASTA data structuring functions | pp. |

**- Peptide data structuring**

Following user selection of runtime parameters, the *IgFamily* program will parse the data .csv file(s). The data are stored on contiguous RAM buffers at runtime. From this data, the raw peptide creates two additional data types for the peptide with modifications removed and the peptide truncated based on associated de novo local confidence scores. Truncation requires a moving average of 85% for amino acid local confidence and a minimum peptide length of 5. In the event of a peptide cleaved at a midpoint such that two peptides of length >= 5 are produced, both are assigned to unique data structures.

|  |  |
| --- | --- |
| Peptide data structuring functions | pp. |

**- External blastp integration**

Following runtime data structuring of FASTA and peptide data, a blastp reference database is created from the current FASTA data, and an input peptide query list is created from the current peptide data. The input peptide query list is tested against the blast reference database and a results file generated as an output file. The blastp output is parsed into runtime and assigned to a homology data structure. Data assigned are the blastp query, the blastp query alignment, the blastp subject, the blastp subject database accession, the blastp query alignment index, the blastp subject alignment index, and the blastp sequence alignment expectation value. Blastp is programmed to allow up to 200 matches for each query, and a generous threshold for alignment acceptance.

|  |  |
| --- | --- |
| Peptide data structuring functions | pp. |

**- Homology data structuring**

Generated and parsed homology data is runtime associated with respective FASTA and peptide data. Additional data types are created -

- Transformed expectation value: The e-value transformed as a metric of relative likelihood.

- Conjugated expectation value: The e-value transformed to consider overall family evidence.

- Homology parameter density: The density of the e-value (0<=par<=1) compared to all

|  |  |
| --- | --- |
| Peptide data structuring functions | pp. |

Narrative

**- *There’s method in models***

“A basic issue with interpreting model-based estimates is in knowing the meaning of parameters. There is no consensus about what a parameter means, however, because different people take different philosophical stances towards models. The perspective in this book is a common Bayesian perspective: Posterior probabilities of parameter values describe the relative compatibility of different states of the world with the data, according to the model. These are small world numbers. Reasonable people may disagree with the large world meaning, and the details of those disagreements depend strongly upon context. Such disagreements are productive, because they lead to model criticism and revision.”

Considerations

**- *A unique peptide not by any other name***

Unique peptides can provide strong support for a gene family and have been used in our lab and among the literature for a range of studies. However, there is a difficulty with unique peptides and immunoglobulins, such that the **uniqueness** or **distinctiveness** of a peptide is not necessarily determined by complete peptide sequence identity.

Consider the IGHV1-69 unique peptide shown in Table X from a truncated homology\_summary output.

Table X - The IGHV1-69 unique peptide FGTANYAQK.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| query | subject | subject\_accession | mismatch\_count | homology | homology\_density\_conjugated |
| FGTANYAQK | FGTANYAQK | IGHV1-69 | 0 | 47 | 22256.4 |
| FGTANYAQK | GNTNYAQK | IGHV1-45 | 3 | 19 | 2112.32 |
| FGTANYAQK | GNTNYAQK | IGHV1-18 | 3 | 19 | 2112.32 |
| FGTANYAQK | GNTNYAQK | IGHV1-58 | 3 | 18 | 1835.31 |
| FGTANYAQK | GGTNYAQK | IGHV1-2 | 3 | 18 | 1835.31 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| query | subject | subject\_accession | mismatch\_count | homology\_density |
| YYVDSVK | YYVDSVK | IGHV3-7 | 0 | 0.472811 |
| YYVDSVK | YYVDSVK | IGHV3-52 | 0 | 0.472811 |
| YYVDSVK | HYVDSVK | IGHV3-16 | 1 | 0.00565536 |
| YYVDSVK | YYADSVK | IGHV3-53 | 1 | 0.00451191 |
| YYVDSVK | YYADSVK | IGHV3-66 | 1 | 0.00451191 |
| YYVDSVK | YYADSVK | IGHV3-23 | 1 | 0.00451191 |
| YYVDSVK | YYADSVK | IGHV3-43 | 1 | 0.00451191 |
| YYVDSVK | YYADSVK | IGHV3-30-5 | 1 | 0.00451191 |
| YYVDSVK | YYADSVK | IGHV3-30 | 1 | 0.00451191 |
| YYVDSVK | YYADSVK | IGHV3-30-3 | 1 | 0.00451191 |
| YYVDSVK | YYADSVK | IGHV3-69-1 | 1 | 0.00451191 |
| YYVDSVK | YYADSVK | IGHV3-33 | 1 | 0.00451191 |
| YYVDSVK | YYADSVK | IGHV3-64 | 1 | 0.00400709 |
| YYVDSVK | YYADSVK | IGHV3-48 | 1 | 0.00400709 |
| YYVDSVK | HYADSVK | IGHV3-35 | 2 | 5.24E-05 |
| YYVDSVK | YADSVK | IGHV3-20 | 2 | 2.61E-05 |
| YYVDSVK | YYADSV | IGHV3-47 | 2 | 2.29E-05 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| query | subject | subject\_accession | mismatch\_count | homology\_density |
| YYADSVK | YYADSVK | IGHV3-23 | 0 | 0.0829693 |
| YYADSVK | YYADSVK | IGHV3-53 | 0 | 0.0829693 |
| YYADSVK | YYADSVK | IGHV3-64 | 0 | 0.0829693 |
| YYADSVK | YYADSVK | IGHV3-48 | 0 | 0.0829693 |
| YYADSVK | YYADSVK | IGHV3-66 | 0 | 0.0829693 |
| YYADSVK | YYADSVK | IGHV3-21 | 0 | 0.0829693 |
| YYADSVK | YYADSVK | IGHV3-43 | 0 | 0.0829693 |
| YYADSVK | YYADSVK | IGHV3-69-1 | 0 | 0.0829693 |
| YYADSVK | YYADSVK | IGHV3-30-5 | 0 | 0.0829693 |
| YYADSVK | YYADSVK | IGHV3-30 | 0 | 0.0829693 |
| YYADSVK | YYADSVK | IGHV3-30-3 | 0 | 0.0829693 |
| YYADSVK | YYADSVK | IGHV3-33 | 0 | 0.0829693 |
| YYADSVK | HYADSVK | IGHV3-35 | 1 | 0.000888038 |
| YYADSVK | YYVDSVK | IGHV3-52 | 1 | 0.000791754 |
| YYADSVK | YADSVK | IGHV3-20 | 1 | 0.000448287 |
| YYADSVK | YYVDSVK | IGHV3-7 | 1 | 0.000703167 |
| YYADSVK | YYADSV | IGHV3-47 | 1 | 0.00039813 |
| YYADSVK | YADSVK | IGHV3-74 | 1 | 0.00039813 |
| YYADSVK | YADSVK | IGHV3-9 | 1 | 0.000352115 |
| YYADSVK | YADSVK | IGHV3-11 | 1 | 0.000352115 |
| YYADSVK | HHADSVK | IGHV3-32 | 2 | 8.10E-06 |
| YYADSVK | HYVDSVK | IGHV3-16 | 2 | 7.10E-06 |
| YYADSVK | HHADSVK | IGHV3-29 | 2 | 7.10E-06 |
| YYADSVK | HHADSVK | IGHV3-30-42 | 2 | 7.10E-06 |
| YYADSVK | HHADSVK | IGHV3-30-22 | 2 | 7.10E-06 |

**- *Distance: It’s all relative***

Gah

Proposals

**- Analysis of data generation and reproduction**

**- Analysis of germline phylogeny**

**- Analysis of germline divergence**

**- Blastp internal integration**

There are notable limitations with using blastp through an external command line. Although parameters are able to be directed to blastp during IgFamily runtime, the modifiable parameters, while thorough, is restricted. In particular, blastp seems to set a hard-coded threshold for score output. This would not be an oversight for simple protein identification, but consider the following homology matches -

**- Blastp custom substitution matrix**

The BLOSUM62 substitution matrix is most often used for defining the rate of amino acid substitution. Specifically it describes the relative likelihood of any amino acid change occurring against an expected query amino acid.



**- Blastp custom conservation weighting**

The BLOSUM62 substitution matrix described above further assumes that the rate of substitution is spatially uniform - That is, any amino acid is equally likely to be substituted than any other. It is known that this is not the case for immunoglobulins, with a relatively greater likelihood of mutation in the hypervariable regions.

**- Advanced statistical modellng**

**- User interface and interaction**

**- Data filesystem**

**- Further automation**

**- De novo and database processes**

The algorithmic differences between de novo and database peptide assignment are suspected to confer some level of error to spectral assignment. It is not known what factors are responsible for incorrect assignment in either case.

Production

Version: v0.9.6

Release: 2016-09-17

Codebase: 4,857 source lines of code

Dependency: 18 files

Version history: 139 commits

Codebase additions: 21,040 source lines of code

Codebase deletions: 16,083 source lines of code

Development environment: Microsoft Visual Studio Community 2015

Version control: Git