

Intel® oneAPI Performance Libraries



Gennady Fedorov, Technical Consulting Engineer, Intel Architecture, Graphics & Software (IAGS)

Intel® oneAPI Base Toolkit

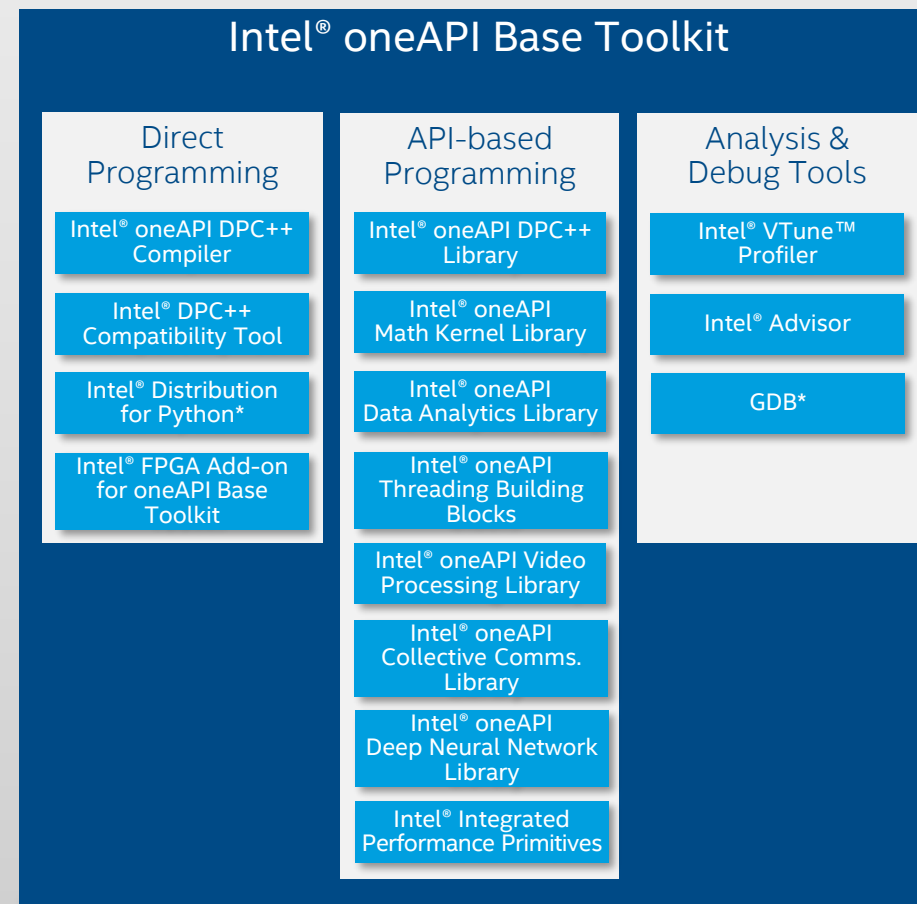
Core set of frequently used tools and libraries for developing high-performance applications across diverse architectures—CPU, GPU, FPGA.

Who Uses It?

- A broad range of developers across industries
- Add-on toolkit users because this is the base for all toolkits

Top Features/Benefits

- Data Parallel C++ (DPC++) compiler, library, and analysis tools
- DPC++ Compatibility tool helps migrate existing CUDA code
- Python distribution includes accelerated scikit-learn, NumPy, SciPy libraries
- Optimized performance libraries for threading, math, data analytics, deep learning, and video/image/signal processing



Intel® oneAPI Base Toolkit

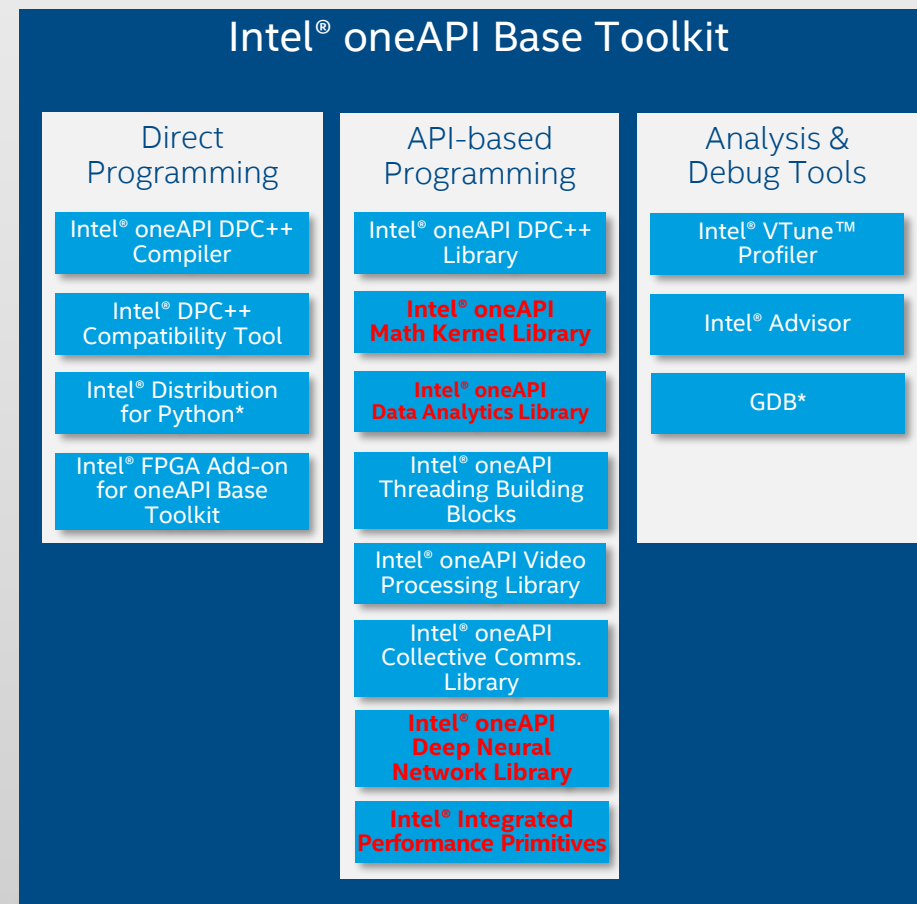
Core set of frequently used tools and libraries for developing high-performance applications across diverse architectures—CPU, GPU, FPGA.

Who Uses It?

- A broad range of developers across industries
- Add-on toolkit users because this is the base for all toolkits

Top Features/Benefits

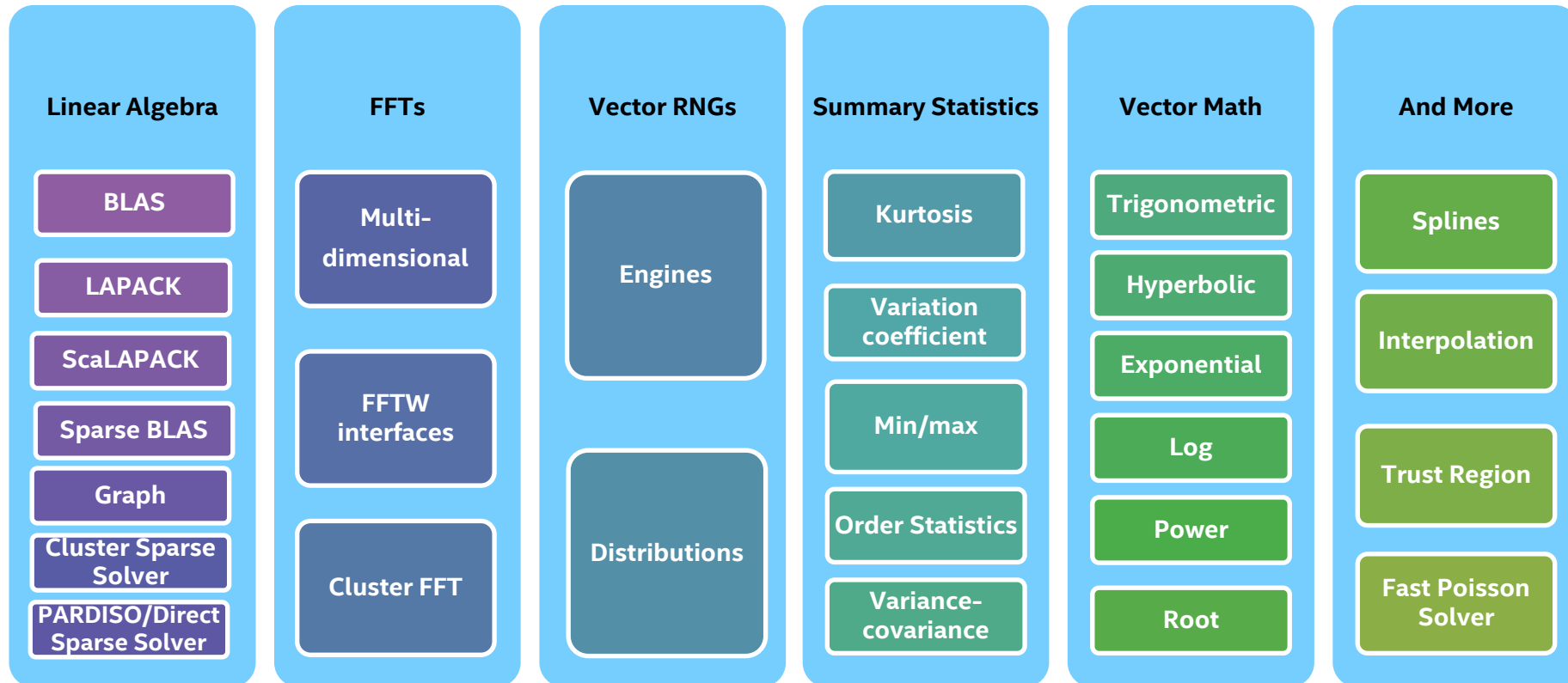
- Data Parallel C++ (DPC++) compiler, library, and analysis tools
- DPC++ Compatibility tool helps migrate existing CUDA code
- Python distribution includes accelerated scikit-learn, NumPy, SciPy libraries
- Optimized performance libraries for threading, math, data analytics, deep learning, and video/image/signal processing



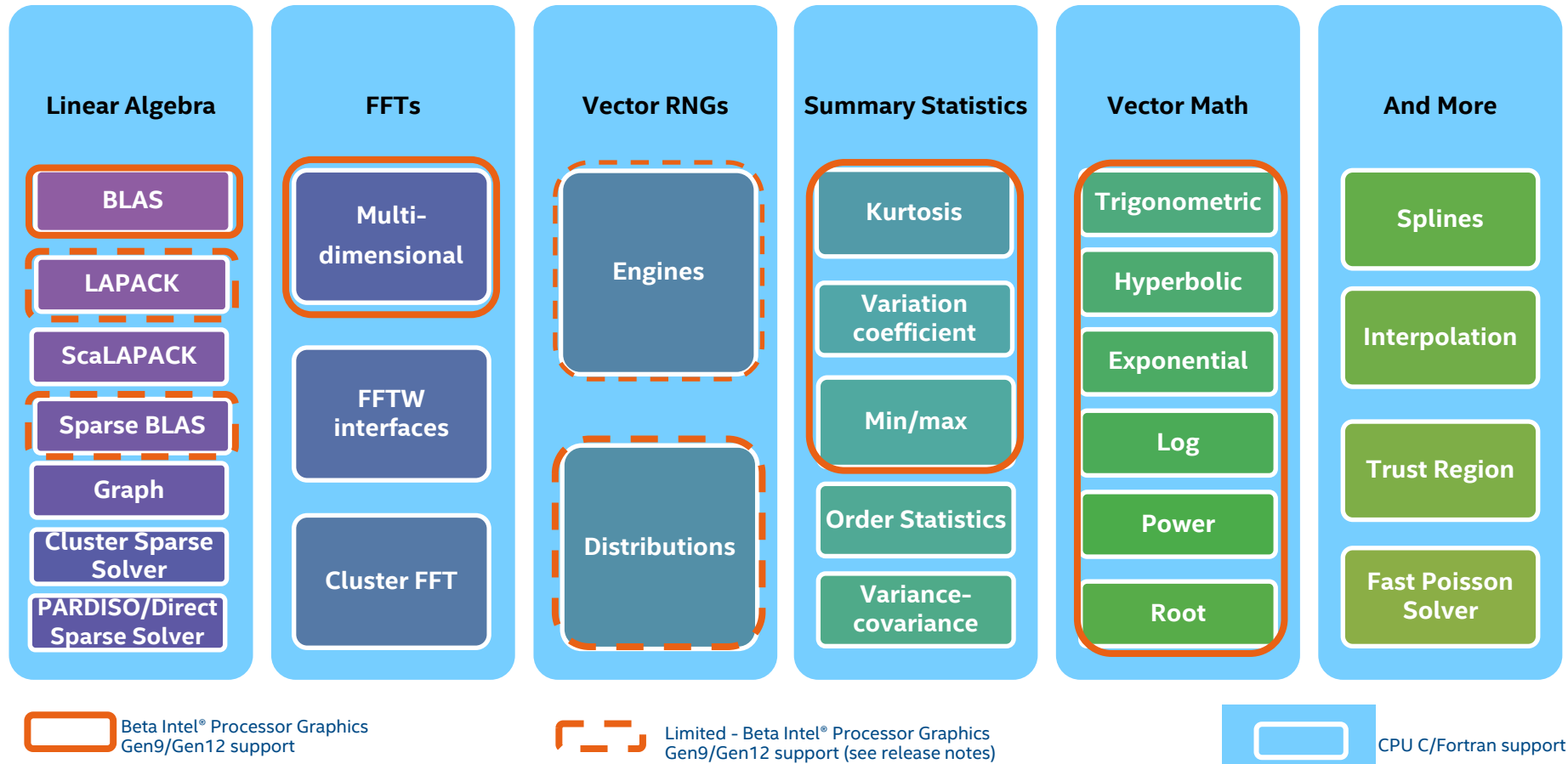
Agenda

- Intel® oneAPI Math Kernel Library (oneMKL)
- Intel® Integrated Performance Primitives (IPP)
- Intel® oneAPI Data Analytics Library (oneDAL)
- References

Intel® oneAPI Math Kernel Library (oneMKL)



Intel® oneAPI Math Kernel Library (oneMKL), cont



Intel® oneAPI MKL, Domain areas

Domain	CPU APIs			Intel GPU APIs		
	DPC++	C	Fortran	DPC++	C OpenMP* Offload	Fortran OpenMP* Offload
BLAS and BLAS-like Extensions	Yes	Yes	Yes	Yes	Yes	Yes
LAPACK and LAPACK-like Extensions	Yes1	Yes	Yes	Yes1	Yes2	Yes2
ScaLAPACK	No	Yes	Yes	No	No	No
Vector Math	Yes	Yes	Yes	Yes	Yes	Yes
Vector Statistics (Random Number Generators)	Yes	Yes	Yes	Yes1	Yes1	Yes1
Vector Statistics (Summary Statistics)	Yes1	Yes	Yes	Yes1	Yes1	Yes1
Data Fitting	No	Yes	Yes	No	No	No
FFT/DFT	Yes	Yes	Yes	Yes4	Yes	Yes
Sparse BLAS	Yes1	Yes	Yes	Yes1	Yes2	No
Sparse Solvers	No	Yes	Yes	No	No	No

1: Subset of the full functionality available. Refer to the [DPC++ developer reference](#) for full list of DPC++ functionality supported.

2: Subset of the full functionality available. For the list of functionality, refer to the developer reference ([C](#) and [Fortran](#))

~~3: Supported on Linux* only.~~

~~4: DFTI interfaces are supported; FFTW interfaces are not supported. (C/F offloading – yes)~~

5: Subset of the full functionality available. Refer to the [DPC++ developer reference](#) for full list of DPC++ functionality supported or to the developer reference for [C](#) and [Fortran](#). Functions which are not implemented for GPU can still be used and will be executed transparently on the host CPU.

Demo - GEMM with oneMKL C API

```
int main() {
    long m = 10, n = 6, k = 8, lda = 12, ldb = 8, ldc = 10;
    long sizea = lda * k, sizeb = ldb * n, sizec = ldc * n;
    double alpha = 1.0, beta = 0.0;

    // Allocate matrices
    double *A = (double *) mkl_malloc(sizeof(double) * sizea, 64);
    double *B = (double *) mkl_malloc(sizeof(double) * sizeb, 64);
    double *C = (double *) mkl_malloc(sizeof(double) * sizec, 64);

    // Initialize matrices [...]
    ...

    // Compute C = A * B on CPU
    cblas_dgemm(CblasColMajor, CblasNoTrans, CblasNoTrans, m, n, k,
                alpha, A, lda, B, ldb, beta, C, ldc);

    ...
}
```

$$C \leftarrow \alpha AB + \beta C$$

Demo - GEMM with oneMKL C OpenMP Offload

```
int main() {
    long m = 10, n = 6, k = 8, lda = 12, ldb = 8, ldc = 10;
    long sizea = lda * k, sizeb = ldb * n, sizec = ldc * n;
    double alpha = 1.0, beta = 0.0;

    // Allocate matrices
    double *A = (double *) mkl_malloc(sizeof(double) * sizea, 64);
    double *B = (double *) mkl_malloc(sizeof(double) * sizeb, 64);
    double *C = (double *) mkl_malloc(sizeof(double) * sizec, 64);

    // Initialize matrices [...]
    ...
    #pragma omp target data map(to:A[0:sizea],B[0:sizeb]) map(tofrom:C[0:sizec])
    {
        #pragma omp target variant dispatch use_device_ptr(A, B, C) nowait
        {
            // Compute C = A * B on GPU
            cblas_dgemm(CblasColMajor, CblasNoTrans, CblasNoTrans, m, n, k,
                        alpha, A, lda, B, ldb, beta, C, ldc);
        }
    }
    ...
}
```

$$C \leftarrow \alpha AB + \beta C$$

Use **target data map** to send matrices to the device

Use **target variant dispatch** to request GPU execution for cblas_dgemm

List mapped device pointers in the **use_device_ptr** clause

Optional **nowait** clause for asynchronous execution
Use **#pragma omp taskwait** for synchronization

Demo - GEMM with oneMKL C API

```
int main() {
```

$$C \leftarrow \alpha AB + \beta C$$

```
    int64_t m = 10, n = 6, k = 8, lda = 12, ldb = 8, ldc = 10;  
    int64_t sizea = lda * k, sizeb = ldb * n, sizec = ldc * n;  
    double alpha = 1.0, beta = 0.0;
```

```
    // Allocate matrices  
    double *A = (double *) mkl_malloc(sizeof(double) * sizea);  
    double *B = (double *) mkl_malloc(sizeof(double) * sizeb);  
    double *C = (double *) mkl_malloc(sizeof(double) * sizec);
```

```
    // Initialize matrices [...]
```

```
    ...  
    cblas_dgemm(CblasColMajor, CblasNoTrans, CblasNoTrans, m, n, k,  
                alpha, A, lda, B, ldb, beta, C, ldc);
```

```
    ...  
}
```

Demo - GEMM with oneMKL DPC++

```
int main() {  
    using namespace oneapi::mkl;  
  
    int64_t m = 10, n = 6, k = 8, lda = 12, ldb = 8, ldc = 10;  
    int64_t sizea = lda * k, sizeb = ldb * n, sizec = ldc * n;  
    double alpha = 1.0, beta = 0.0;  
  
    sycl::queue Q{sycl::gpu_selector{}};  
  
    // Allocate matrices  
    double *A = malloc_shared<double>(sizea, Q);  
    double *B = malloc_shared<double>(sizeb, Q);  
    double *C = malloc_shared<double>(sizec, Q);  
  
    // Initialize matrices [...]  
    ...  
  
    auto e = blas::gemm(Q, transpose::N, transpose::N, m, n, k,  
                       alpha, A, lda, B, ldb, beta, C, ldc));  
    ...  
}
```

$$C \leftarrow \alpha AB + \beta C$$

Set up GPU queue

Allocate CPU/GPU-accessible shared memory

New oneMKL DPC++ API
Computation is performed
on given queue

Output **e** is a sycl::event object representing command completion
Call **e.wait()** to wait for completion

Intel® Integrated Performance Primitives (IPP)

What is Intel® IPP?

Intel IPP provides developers with ready-to-use, processor-optimized functions to accelerate **Image & Signal processing, Data Compression & Cryptography computation tasks**

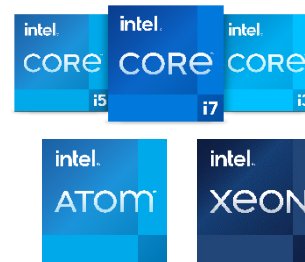
Why use Intel® IPP?

- High Performance
- Easy to Use APIs
- Faster Time to Market (TTM)
- Production Ready
- Cross-Platform API

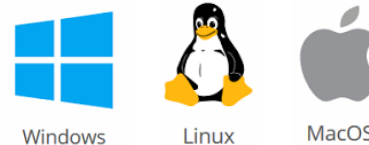
How to get Intel® IPP?

- [Intel® oneAPI Base Toolkit](#)
- [Free Tools Program](#)

Optimized for



Supports



Addresses

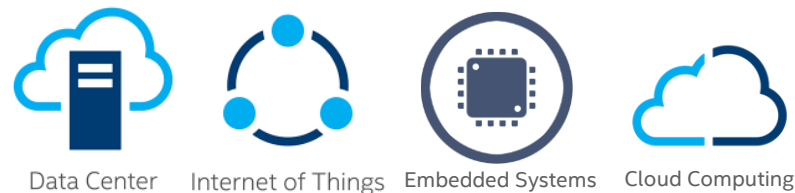


Image Processing Uses

- Medical Imaging
- Computer Vision
- Digital Surveillance
- ADAS
- Automated Sorting
- Biometric Identification
- Visual Search

Signal Processing Uses

- Games (audio control or effects)
- Echo Cancellation
- Telecommunication
- Energy

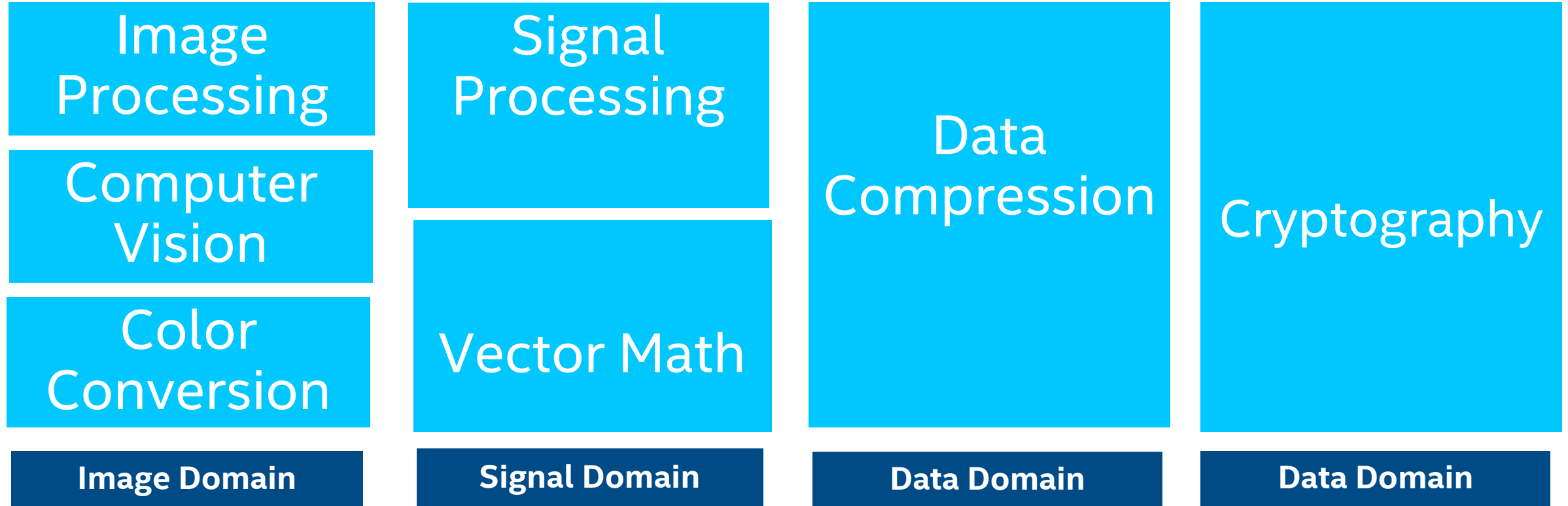
Data Compression & Cryptography Uses

- Data centers
- Enterprise data management
- ID Verification
- Smart Cards/wallets
- Electronic Signature
- Information Security/Cybersecurity

Find out more at: <http://software.intel.com/intel-ipp>

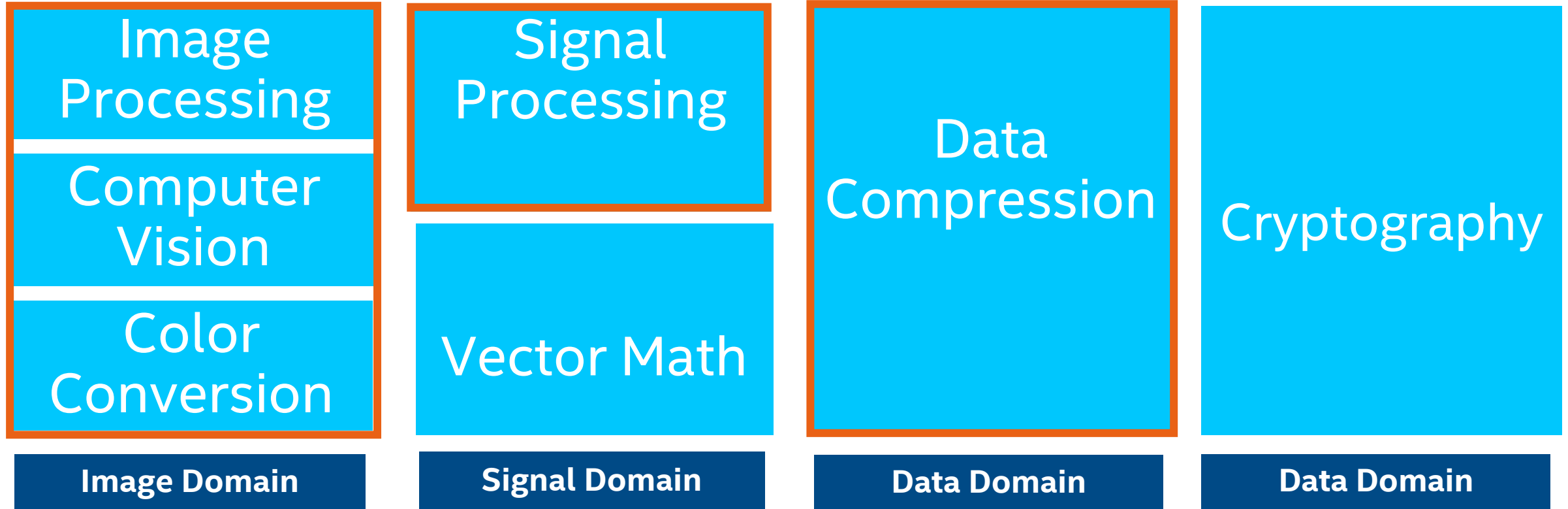
What's Inside Intel® Integrated Performance Primitives

High Performance , Easy-to-Use & Production Ready APIs



What's Inside Intel® Integrated Performance Primitives

High Performance , Easy-to-Use & Production Ready APIs



Beta Intel® Processor Graphics Gen9/Gen12 support

oneIPL Specification - <https://spec.oneapi.io/oneipl/latest/index.html>

oneDTL Specification - <https://spec.oneapi.io/onedtl/latest/index.html>

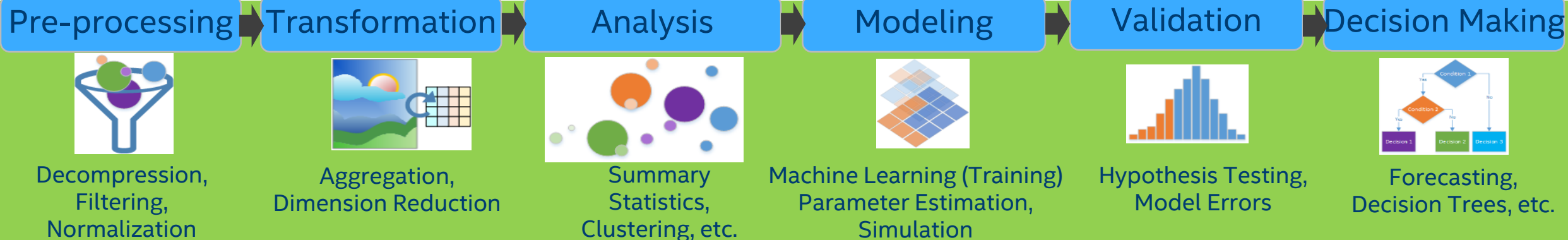
Intel® oneAPI Data Analytics Library (oneDAL)

- Boost Machine Learning & Data Analytics Performance
- Helps applications deliver better predictions faster
- Optimizes data ingestion & algorithmic compute together for highest performance
- Supports offline, streaming & distributed usage models to meet a range of application needs
- Split analytics workloads between edge devices and cloud to optimize overall application throughput

Learn More:
software.intel.com/oneAPI/oneDAL

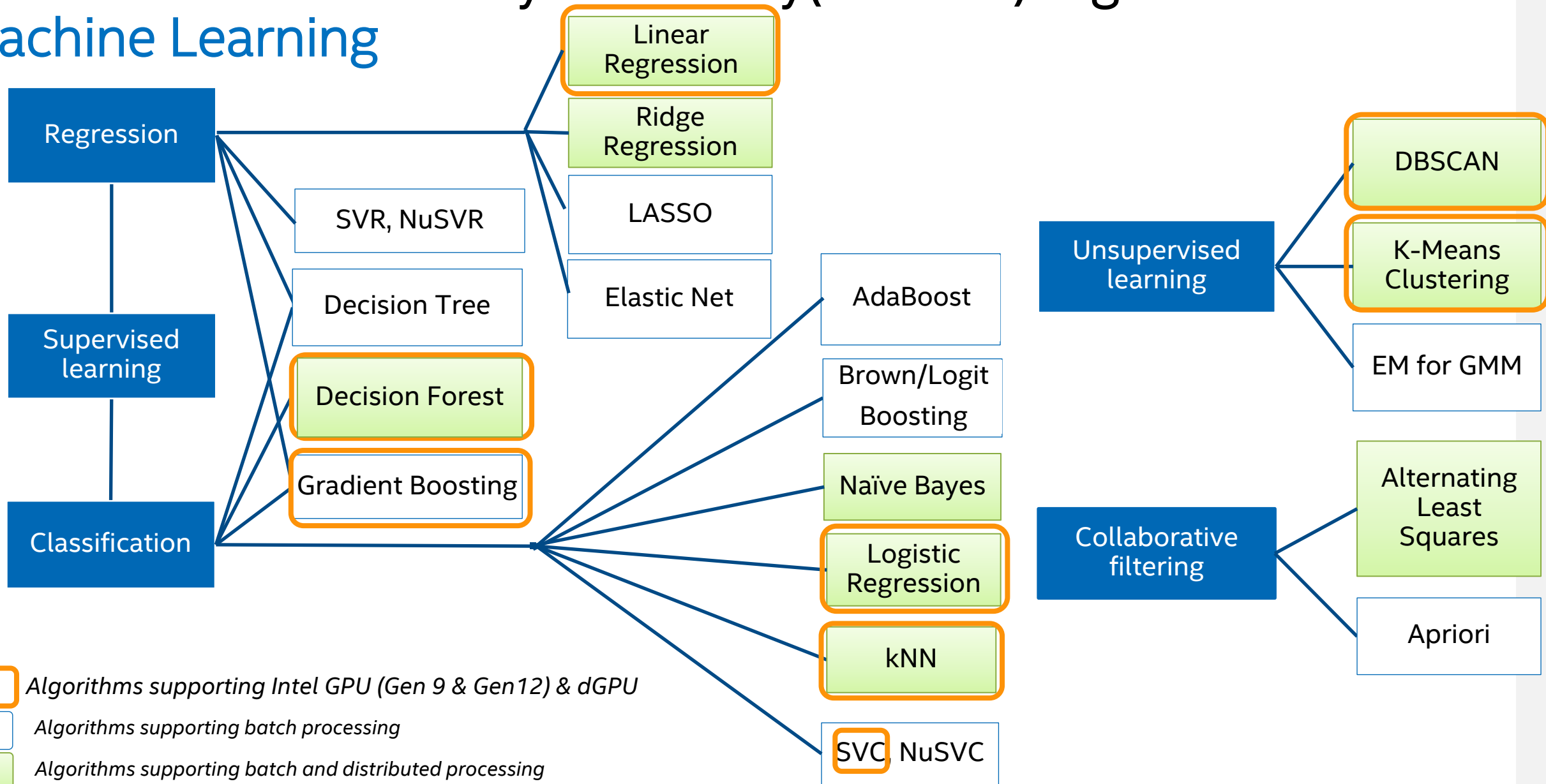
What's New in the oneDAL Release (2021.2-2021.5)

- Intel® Extension for Scikit-learn
 - Support **python 3.6-3.9** versions
 - Download from **pip** and **conda-forge** channel
- New CPU support for the following:
SVR, NuSVMs, Multiclass SVM, KNN brute force
- DPC++ GPU support for the following:
 - Batch: **K-means , Covariance, Basics statistics, PCA, KNN, SVC, DBSCAN**
 - Distributed: **Covariance, DBSCAN, Decision Forest, Basics statistics**



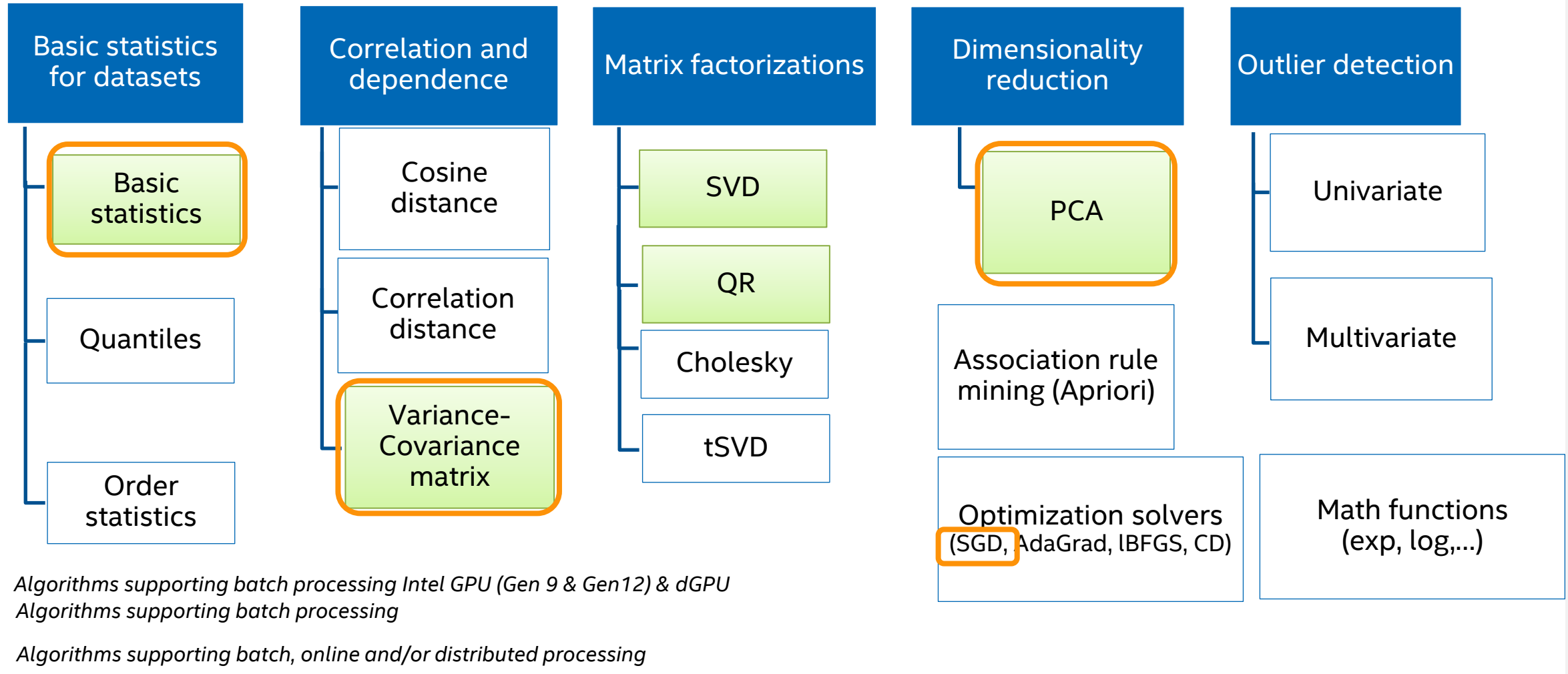
Intel® oneAPI Data Analytics Library(oneDAL) Algorithms

Machine Learning



Intel® oneAPI Data Analytics Library (oneDAL) algorithms

Data Transformation and Analysis



Intel® oneMKL Resources

Intel® oneMKL Product Page	https://www.intel.com/content/www/us/en/developer/tools/oneapi/onemkl.html		
Get Started with Intel® oneMKL	https://www.intel.com/content/www/us/en/develop/documentation/get-started-with-mkl-for-dpcpp/top.html		
Intel® oneMKL Developer Reference	https://www.intel.com/content/www/us/en/develop/documentation/onemkl-developer-reference-c/top.html		
Intel® oneMKL Developer Guide	https://www.intel.com/content/www/us/en/develop/documentation/onemkl-windows-developer-guide/top.html		
Intel® oneMKL Specification	https://spec.oneapi.io/versions/latest/elements/oneMKL/source/index.html		
Intel® oneMKL Open-Source Interface	https://github.com/oneapi-src/oneMKL		
Intel® oneMKL Release Notes	https://cqpreview.intel.com/content/www/us/en/developer/articles/release-notes/onemkl-release-notes.html		
Intel® oneMKL Forum	https://community.intel.com/t5/Intel-oneAPI-Math-Kernel-Library/bd-p/oneapi-math-kernel-library		

Intel® oneDAL Resources

Intel® oneDAL Product Page	https://www.intel.com/content/www/us/en/develop/documentation/oneapi-programming-guide/top/api-based-programming/intel-oneapi-data-analytics-library-onedal.html									
Intel® oneDAL Release Note	https://www.intel.com/content/www/us/en/developer/articles/release-notes/oneapi-dal-release-notes.html									
Intel® oneDAL Forum	https://community.intel.com/t5/Intel-oneAPI-Data-Analytics/bd-p/oneapi-data-analytics-library									
Get Started with Intel® oneDAL	https://www.intel.com/content/www/us/en/develop/documentation/get-started-intel-oneapi-data-analytics-library/top.html									
Intel® oneDAL Documentation	https://oneapi-src.github.io/oneDAL/									

Intel® IPP Resources

Intel® IPP Product Page	https://www.intel.com/content/www/us/en/developer/tools/oneapi/ipp.html					
Intel® IPP Release Notes	https://www.intel.com/content/www/us/en/developer/articles/release-notes/release-notes-for-oneapi-integrated-performance-c					
Intel® IPP Forum	https://community.intel.com/t5/Intel-Integrated-Performance/bd-p/integrated-performance-primitives					
Get Started with Intel® IPP	https://www.intel.com/content/www/us/en/develop/documentation/get-started-with-ipp-for-oneapi-windows/top.html					
Intel® IPP Developer Reference	https://www.intel.com/content/www/us/en/develop/documentation/ipp-dev-reference/top.html					
Intel® IPP Developer Guide	https://www.intel.com/content/www/us/en/develop/documentation/dev-guide-ipp-for-oneapi/top.html					
Intel® oneAPI Toolkit Products v	https://www.intel.com/content/www/us/en/developer/tools/oneapi/support.html					

Notices & Disclaimers

Intel technologies may require enabled hardware, software or service activation. Learn more at intel.com or from the OEM or retailer.

Your costs and results may vary.

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.

Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice Revision #20110804. <https://software.intel.com/en-us/articles/optimization-notice>

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors.

Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. See backup for configuration details. For more complete information about performance and benchmark results, visit www.intel.com/benchmarks.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See configuration disclosure for details. No product or component can be absolutely secure.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

© Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.

